НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК_____

«До захисту допущено»
Завідувач кафедри
_____ О.В. Коваль
(підпис) (ініціали, прізвище)
“___”_____2019р.

Магістерська дисертація

зі спеціальності - 122 Комп’ютерні науки
за спеціалізацією - Комп'ютерний моніторинг та геометричне моделювання процесів і систем
на тему: Розробка системи підтримки прийняття рішень при аналізі інформаційного забезпечення безпеки даних енергетичних процесів та систем. Підсистема збереження даних

Виконав (-ла): студент (-ка) 6 курсу, групи ТМ-81мп
Кудряшова Ольга Борисівна _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник к.т.н., доцент Ходаковський О.В. _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2019

Національний технічний університет України
“ Київський політехнічний інститут ім. Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 122 Комп’ютерні науки
за спеціалізацією - Комп'ютерний моніторинг та геометричне моделювання процесів і систем

ЗАТВЕРДЖУЮ
Завідувач кафедри
Коваль О.В. _____
(прізвище, ініціали) (підпис)
«____» _____ 2019р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Кудряшова Ольга Борисівна
(прізвище, ім’я, по батькові)

1. Тема дисертації Розробка системи підтримки прийняття рішень при аналізі інформаційного забезпечення безпеки даних енергетичних процесів та систем. Підсистема збереження даних

Науковий керівник к.т.н., доцент Ходаковський О.В.
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “4” листопада 2019 року №3812-с

2. Строк подання студентом дисертації “ ” 201 року

3. Об’єкт дослідження система підтримки прийняття рішень при інформаційному забезпеченні безпеки даних енергетичних процесів та систем

4. Предмет дослідження система із забезпечення безпеки даних.

5. Перелік питань, які потрібно розробити
1) проаналізувати сучасні методи щодо забезпечення безпеки даних;
2) проаналізувати існуючі системи щодо збереження конфіденційних даних;
3) розробити структуру для побудови системи;
4) розробити користувацький інтерфейс;
5) розробити програмне забезпечення.

6. Орієнтований перелік ілюстративного матеріалу актуальність, мета роботи, завдання та методи досліджень, системи-аналоги, база даних, архітектура програмного

продукту, початок роботи, аналіз графіків, результати виконання програми, висновки

7. Орієнтований перелік публікацій _____

_____

8. Дата видачі завдання «14» січня 2019 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання		
2	Збір інформації		
3	Аналіз вимог завдання, вибір методів і засобів розв'язання поставленої задачі		
4	Підготовка матеріалів магістерської роботи		
5	Проміжний контроль підготовки		
6	Підготовка публікацій		
7	Підготовка доповідей на конференціях за темою магістерської роботи		
8	Доповідь на конференції		
9	Написання основних розділів автореферату		
10	Звіт за перший рік роботи над магістерською дисертацією		

Студент _____ Кудряшова О.Б.
(підпис) (прізвище та ініціали)

Науковий керівник _____ Ходаковський О.В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Структура й обсяг дипломної роботи. Магістерська дисертація складається зі вступу, 5 розділів, висновку, переліку посилань з 19 найменуваннь, і містить 23 рисунків, 16 таблиць. Повний обсяг магістерської дисертації складає 85 сторінок, з яких перелік посилань займає 2 сторінки.

Актуальність теми. В сучасному світі є дуже актуальною проблемою збереженість даних та конфіденційної інформації, і особливо інформації, що зберігається у базах даних. Наразі всі найбільші корпорації світу у сфері інформаційних технологій намагаються вберігти свої конфіденційні дані від зловмисників. А також зберігти їх цілісність, доступність, оприлюднення тощо. Найбільш актуальною ця проблема є для персональних даних, наприклад, даних про користувачів або даних про банківські транзакції. Розвиток інформаційних технологій і їх імплементація майже в усі сфери життя є на сьогодні однією з головних причин для створення і проектування систем із забезпечення захисту даних. Це спричинило появу систем для збереження безпеки даних, а також появу вбудованих функцій і можливостей захисту даних в сучасних СКБД.

Мета дослідження полягає в створенні засобів та інструментів для збереження конфіденційної інформації і даних.

Об'єктом дослідження є можливість системи аналізувати спроби несанкціонованого доступу до баз даних та видача користувачу рекомендацій відповідно до зробленого аналізу загроз.

Предметом дослідження є комп'ютерне програмне забезпечення із забезпечення безпеки даних, аналіз несанкціонованого доступу до баз даних, збереження баз даних для уникнення модифікації або знищення даних у них.

Наукова новизна одержаних результатів. Найбільш суттєвими науковими результатами магістерської дисертації є:

- збереження баз даних в залежності від типу бази даних, а також від аналізу загроз, тобто спроб несанкціонованого доступу;
- видача користувачу рекомендацій щодо збереження даних, в залежності

від проведеного аналізу загроз.

Практичне значення. Створений програмний продукт, метою якого збереження даних при спробах несанкціонованого доступу до них та видача рекомендацій користувачу (наприклад, адміністратору баз даних) для їх усунення, забезпечує більш якісний контроль за збереженням цілісності даних та інформації, та може бути використаний у будь-яких сферах та компаніях, які потребують максимально надійної захищеності конфіденційної інформації.

ABSTRACT

Structure and volume of thesis. The master's dissertation consists of an introduction, 5 sections, conclusion, a list of references from 19 titles, and contains 23 drawings, 16 tables. The full volume of the master's dissertation is 85 pages, of which the list of links takes 2 pages.

Actuality of theme. In today's world, data and confidential information, and especially information stored in databases, is a very relevant issue. Nowadays, all of the world's largest IT corporations are trying to keep their sensitive information safe from abusers. Also they are trying to preserve their integrity, accessibility, publicity and more. This issue is most relevant for personal data, such as user data or bank transaction data. The development of information technologies and their implementation in almost all spheres of life is one of the main reasons for creating and designing data protection systems. This has led to the emergence of systems for maintaining data security, as well as the emergence of built-in features and data protection capabilities in modern DBMS.

The purpose of the study to create tools and methods to store sensitive information and data.

The object of research is the ability of the system to analyze the attempts of unauthorized access to databases and giving recommendations to the user according to the analysis of threats.

The subject of the research is a computer software for data security, analysis of unauthorized access to databases, storing of databases to avoid modification or destruction of data in them.

Scientific novelty of the obtained results. The most significant scientific results of the master's thesis are:

- storage of databases depending on the type of a database, as well as on the analysis of threats, ie attempts of unauthorized access;
- giving recommendations to the user about saving the data, depending on the analysis of threats.

Practical meaning. A software product designed to preserve data from attempts to gain unauthorized access and provide recommendations to the user (such as a database administrator) to eliminate them, provides better control over the integrity of the data and information, and can be used in any area and by all the companies that need the most secure protection of sensitive information.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

БД – база даних

СКДБ – Система керування базами даних

IL – Intermediate Language

JIT – Just in time, тип компілятора

SQL – Structured query language, мова структурованих запитів

CLR – Common Language Runtime, віртуальна машина для виконання коду на мовах .NET

CLI – Common Language Infrastructure, специфікація загальномовної інфраструктури

ОС – операційна система

IDE– Integrated Development Environment, середовище розробки

ВСТУП

В сучасному світі практично кожна сучасна компанія не може обійтися без використання баз даних в своїй діяльності. Бази даних (БД) – дуже важливий і цінний актив для будь-якої компанії. Оскільки в базі даних можуть зберігатися персональні або конфіденційні дані, варто дуже відповідально віднестися до їх захисту. Будь-які несправності в роботі СКБД і баз даних спрчинити катастрофічі наслідки. Згідно з результатами сучасних досліджень, грошові збитки від порушення одного з властивостей «конфіденційність-цілісність-доступність» запису бази даних складає від $ 100 до $ 240. Видатки виділяються на відновлення втрачених даних, розслідування факту втрати даних, відновлення та ліквідацію збитків іміджу компанії і т.д. Тому проблема забезпечення безпеки баз даних є дуже актуальною[1].

Під «захистом БД» мається на увазі різні способи для запобігання несанкціонованому доступу до інформації, що зберігається в таблицях. Одним з найбільш вразливих місць при забезпеченні безпеки даних (захист конфіденційних даних), як правило, є занадто велика кількість людей, які можуть отримати доступ до них на різних рівнях. Тобто загрози щодо збереження в базах даних інформації можуть виникнути не тільки ззовні, але і зсередини з боку легальних користувачів. Найбільш типовим прикладом є скачування бази даних системним адміністратором перед звільненням, або злодійяння щодо бази даних співробітником, які мають до неї доступ у зв'язку з посадовими обов'язками. Таким чином, незалежно від рівня захищеності каналів доступу до інформації, поки існують папір і ручка, не можна бути абсолютно впевненими, що безпека баз даних відповідає корпоративним вимогам.

У фінансовому секторі в базах даних зберігається інформація про клієнтів, їх рахунках, грошових транзакціях, а, наприклад, в секторі нафтогазовидобування – показники про видобуток, зберігання, транспортування і відвантаження продуктів нафти і газу.

Зростаюча роль інформації в житті сучасних компаній привела до вибухового зростання обсягів даних. Більше половини з них зберігаються в форматі баз даних. І дуже часто в загальній масі присутні в тому числі і «критична» інформація. У фінансовому секторі в базах даних зберігається інформація про клієнтів, їх рахунках, грошових транзакціях, в нафтогазовому секторі – показники видобутку, транспортування, зберігання і відвантаження нафтопродуктів.

Атаки на сховища і БД є одними з найнебезпечніших і найзатратніших для підприємств і організацій. Згідно зі статистикою компанії infowatch, в останні роки кількість витоків даних в світі невпинно зростає, при цьому на 2015 рік понад тридцять відсотків з них припадають на зовнішніх порушників і більш як шістдесят виконано за участі співробітників компаній. Навіть якщо зробити припущення, що в певних випадках витік включав дані, до яких співробітник має право доступу, лише кожен третій випадок виявлявся зовнішньою атакою. Також варто зазначити, що, згідно з наведеним даними, звнішніми атаками є сім з восьми витоків згальним обсягом понад десять мільйонів записів.

Зловмисники зазвичай прагнуть отримати доступ до таких видів інформації, як внутрішня корпортативна інформація, персональні дані співробітників, фінансова інформація, інформація про замовників/клієнтів, інтелектуальна власність, дослідження ринку/аналіз діяльності конкурентів, банківська та транзакційна інформація [2]. Ця інформація зазвичай зберігається в корпоративних сховищах фірм і базах даних різного обсягу.

Всі ці фактори свідчать про необхідність забезпечення захисту не тільки комунікацій, операційних систем та інших елементів інфраструктури, а й баз даних як ще однієї з перешкод на шляху хакера. Проте в сучасному світі робота в сфері забезпечення безпеки БД зазавичай сфокусована на подоланні існуючих і вже відомих вразливостей, реалізацію основних моделей доступу і розгляд питань, специфічних для конкретної СКБД.

Майже всі великі виробники СКБД обмежуються розвитком концепції конфіденційності, цілісності і доступності даних, а їхні дії спрямовані, в основному, на подолання існуючих і вже відомих вразливостей, реалізацію основних моделей доступу і розгляд проблем, що стосуються конкретної СКБД. Такий підхід забезпечує вирішення конкретних завдань, але не сприяє появі і розробці загальної концепції безпеки для такого класу ПЗ, як СКБД. Це значним чином ускладнює завдання щодо забезпечення безпеки сховищ даних на підприємствах.

Найголовніший фактор успішного захисту БД – це знання того, які дані потребують захисту (інтелектуальна власність, фінансова інформація, дані про кредитні картки, кадрові або персональні дані) і як найліпшим чином захистити їх від можливих типів загроз. Для розробки процесу забезпечення безпеки БД необхідно розуміти чинні стандарти, такі як PCI DSS, СТО БР Іббсе-1.0 – 2006, ФЗ «Про персональні дані», закон Сарбейнса-Окслі, Basel II і ін. Природно, дії щодо політики безпеки БД повинні бути інтегровані з загальним процесом забезпечення інформаційної безпеки корпоративної мережі.

Таким чином, сучасні дослідження в області безпеки систем управління базами даних обмежуються розвитком і покращенням концепції конфіденційності, доступності і цілісності даних, що не відповідає вимогам сучасного світу щодо систем захисту та інформаційної безпеки програмного забезепечення, до того ж в контексті конкретних методів захисту, а не загального розгляду проблеми. При цьому вони часто стосуються конкретних програмних продуктів, а не всього класу відповідного програмного забезпечення.

1. ПОСТАНОВКА ЗАДАЧІ

Зазвичай сховища даних в сучасних компаніях складаються з двох компонентів: даних, що зберігаються (власне, БД) і програмного забезпечення для управління та адміністрування (СКБД).

Забезпечення безпеки неможливе, якщо не створити необхідні умови для безпечного менеджменту збереженими даними. Це означає, що всі питання щодо захисту СКБД можна розгалужити на 2 категорії: незалежні і залежні від даних.

Ті вразливості, які від даних не залежать, є характерними і для інших типів програмних продуктів. Причина проблем є різною – це і нерегулярне оновлення, і недостатній досвід або навички системного адміністратора, або наявність функцій, якими не користуються.

Водночас практика демонструє, що більшість аспектів безпеки СКБД в більшій мірі залежить від даних. До прикладу, деякі СКБД забезпечують можливість написання запитів через певні мови, які мають набори методів, доступних для користувача. Архітектура застосованих мов пов'язана з моделлю даних, яка використовується для зберігання даних. Підсумовуючи, можна зробити висновок, що модель частково визначає особливості мови, а мови визначають присутність в ньому деяких вразливостей. При цьому такі вразливості, що часто застосовуються, наприклад, як ін'єкції, виконуються різними шляхами (Java-ін'єкція, SQL-ін'єкція), враховуючи особливості мовного синтаксису.

Метою розробки є створення програмного продукту, який забезпечує можливості збереження конфіденційних та персональних даних при спробах несанкціонованого доступу до них. Окрім цього метою також є аналіз сучасного програмного забезпечення та засобів, які виконують поставлену задачу, а саме, забезпечення безпеки та конфіденційності даних, що зберігаються в БД. Варто також зіставити їх з реальними проблемами сучасних компаній, що працюють з великими обсягами даних та потребують гарантій їх неушкодженості та надійності, а також , при необхідності, адаптувати їх зберігаючи якість роботи на високому рівні.

Для наочної демонстрації роботи системи був розроблений інтерфейс користувача, який дозволяє переглянути всі доступні бази даних для роботи і, за необхідності, виконати певні дії з ними для гарантії збереження цілісності даних у них.

Програмний продукт повинен забезпечувати:

— авторизацію користувача;
— можливість роботи з базою даних;
— можливість вибору бази даних для роботи;
— задання параметрів збереження бази даних;
— аналіз можливих загроз конфіденційній інформації;
— видача рекомендацій користувачеві.

Система повинна відловлювати помилки та виводити повідомлення для користувача у разі виникнення несподіваних умов роботи.

Для подальшого доповнення розробки і оновлення, програмний код системи має бути якнайкраще оптимізований і організований. Програма має бути оптимізована і стабільно працювати.

2. АНАЛІЗ ПРОБЛЕМИ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ДАНИХ. ПІДСИСТЕМА ЗБЕРЕЖЕННЯ ДАНИХ

В цій системі захист інформації та даних передбачає систему заходів та рекомендацій, метою яких є обмеження та запобігання несанкціонованого доступу до конфіденційних даних, несанкціонованого їх спотворення, видалення, порушення цілісності, а також аналіз можливих атак на систему.

2.1 Основні терміни та визначення

З урахуванням визначених принципів пропонуються основні поняття і визначення пов’язані із безпекою даних (таблиця 2.1).

Таблиця 2.1. Основні терміни і визначення

Термін	Визначення
База даних	Сукупність даних, які організовані спеціальним чином, і відображають стан об’єктів та їх зв’язки між один одним відповідно до заданої предметної області.
Система управління базами даних	Набір даних та спеціального програмного забезпечення, що забезпечує функції створення, видалення, збереження, оновлення та пошуку інформації в базах даних з різними рівнями контролю доступу до даних багатьма користувачами.
Мережевий протокол	Набір правил, заснований на певних стандартах, що визначає правила та принципи взаємодії в комп’ютерній мережі. Протокол формує загальні правила взаємодії різних програм, систем чи мережевих вузлів і таким чином створює єдиний простір передачі даних.

Таблиця 2.1 (продовження)

Хост	Сервер (потужний комп'ютер або мобільний комп'ютерний комплекс), який займається розподілом інформації
Порт	Натуральне число, яке записується в заголовка протоколів (TCP, UDP, SCTP, FTP). Використовується для визначення процесу-одержувача пакета в межах одного хоста.
Контроль доступу	Функція системи, що забезпечує певні технології та способи, які дозволяють або забороняють доступ до деяких типів даних, що заснований на виявленні об'єкта, якому потрібен доступ, і даних до яких об’єкт хоче отримати доступ.
Вразливість	Можливість виникнення на якомусь етапі життєвого циклу комп’ютерної системи такого її стану, при якому з’являються умови, що загрожують безпеці інформації чи даних .
Резервна копія БД	Копія даних, яку можна використати для відновлення втрачених даних у разі виникнення помилок. Резервні копії баз даних також використовуються для відновлення копії бази даних в новому розташуванні.
SQL-ін’єкція	Атака, що спрямовується на додаток,в якій створюється SQL-вираз з користувацького вводу шляхом звичайної конкантенації. У разі успішного завершення зловмисник може змінити логіку виконання SQL-запиту так, як йому треба. Частіше за все він виконує звичайний fingerprinting СКБД, а також витягує таблиці з найбільш "цікавими" іменами. Після цього, в залежності від привілеїв, з якими запущено додаток із вразливостями, він може звернутися до захищених частин бекенду веб-застосунку (наприклад, прочитати файли на стороні хоста або виконати інші довільні команди).

Таблиця 2.1 (продовження)

DDoS атака	Атака на певну систему, яка одночасно виконується з великої кількості комп'ютерів для якнайбільшого навантаження на сервер. Мета атаки: перевантажити сервер так, щоб реальні користувачі не могли отримати доступ до певних ресурсів. Для успішної DDoS-атаки зловмисник має посилати більше запитів, ніж може обробити сервер, що атакується. Під час атаки на мережеву частину сервера злодій намагається перевантажити канал зв'язку. Канал зв'язку відповідає за обсяг даних, який сервер може обробити і прийняти. Якщо даних забагато, сервер не може їх обробити і частина відвідувачів не може потрапити на сайт. Під час атаки на програмну частину злодій намагається навантажити якусь із частин сервера: оперативну пам'ять, потужність процесора, граничну кількість процесів або підключень до БД. Коли який-небудь з ресурсів закінчується, сервер зависає і починає гальмувати.
Хеш	Число, яке генерується з заданого тексту за допомогою певного хеш-алгоритму. Це число менше оригінального тексту. Алгоритм працює таким чином, що для кожного тексту генерується унікальний хеш. І відновлення тексту з перехопленого зловмисником хешу, є майже неможливим Одна з незамінних властивостей хешування – його унікальність. Однакові значення хешу можуть бути використані для різного тексту. Навіть найменша зміна в тексті призведе до повного змінення значення хешу. Це ще називають ефектом лавини.

2.2 Сучасні проблеми забезпечення безпеки та способи захисту баз даних

Історично розвиток систем безпеки баз даних відбувався як реакція на різноманітні дії зловмисників. Крім цього, ці важливі зміни також були обумовлені загальним розвитком систем баз даних від рішень на мейнфреймах до хмарних сховищ.

Можна виділити наступні архітектурні підходи:

— повний доступ всіх користувачів до сервера БД;

— поділ користувачів на довірених і частково довірених засобами СКБД;

— введення системи аудиту (логів дій користувачів) засобами СКБД;

— введення шифрування даних; винос коштів аутентифікації за межі СКБД в операційні системи і проміжне ПО; відмова від повністю довіреної адміністратора даних.

Впровадження засобів захисту як реакції на вже відомі загрози не забезпечує захист від нових способів атак і формує розрізнене уявлення про саму проблему забезпечення безпеки.

З урахуванням різних еволюційних особливостей з’явилася і існує велика кількість дуже різних засобів забезпечення безпеки, що в підсумку призвело до відсутності комплексного розуміння безпеки даних. Відсутній загальний підхід до безпеки сховищ даних. Крім того, ускладнюється і прогнозування майбутніх атак, а також розробка механізмів захисту. Більш того, для багатьох систем зберігається актуальність вже давно відомих атак, ускладнюється підготовка фахівців з безпеки[2].

Список основних вразливостей СКБД не зазнав істотних змін за останній час. Проаналізувавши засоби забезпечення безпеки СКБД, архітектуру БД, відомі вразливості і інциденти безпеки, можна виділити наступні причини виникнення такої ситуації:

— проблемами безпеки серйозно займаються тільки великі виробники;

— програмісти баз даних, прикладні програмісти і адміністратори не приділяють належної уваги питанням безпеки;

— різні масштаби і види збережених даних вимагають різних підходів до безпеки;

— різні СКБД використовують різні мовні конструкції для доступу до даних, організованих на основі тієї ж самої моделі;

— з’являються нові види і моделі зберігання даних.

Багато вразливостей зберігають актуальність за рахунок неуваги або незнання адміністраторами систем баз даних питань безпеки. Наприклад, прості SQL-ін’єкції широко експлуатуються сьогодні по відношенню до різних web-додатків, в яких не приділяється достатньо уваги до вхідних даних запитів.

Застосування різних засобів забезпечення інформаційної безпеки є для організації компромісом у фінансовому плані: впровадження більш захищених продуктів і підбір більш кваліфікованого персоналу вимагають великих витрат. Компоненти безпеки часто можуть негативно впливати на продуктивність СКБД[3].

Ці проблеми посилюються з появою і все більшим поширенням нереляційних СКБД, що оперують іншою моделлю даних, однак побудованої за тими ж принципами, що і реляційні. Різноманіття сучасних NoSQL-рішень призводить до різноманітності застосовуваних моделей даних і розмиває межу поняття БД.

Наслідком цих проблем і відсутності єдиних методик є нинішня ситуація з безпекою NoSQL-систем. У більшості NoSQL-систем відсутні не тільки загальноприйняті механізми безпеки на кшталт шифрування, підтримки цілісності та аудиту даних, але навіть розвинені засоби аутентифікації користувачів[4].

На підставі поділу вразливостей можна виділити залежні і незалежні від даних заходів забезпечення безпеки сховищ інформації.

Незалежними від даних можна назвати наступні вимоги до безпечної системі БД:

— функціонування в довіреному середовищі. Під довіреним середовищем слід розуміти інфраструктуру підприємства і його захисні механізми, обумовлені

політикою безпеки. Таким чином, мова йде про функціонування СКБД відповідно до правил безпеки, що застосовуються і до всіх інших систем підприємства;

— організація фізичної безпеки файлів даних. Вимоги до фізичної безпеки файлів даних СКБД в цілому не відрізняються від вимог, що застосовуються до будь-яких інших файлів користувачів і додатків. Різні масштаби і види збережених даних вимагають різних підходів до безпеки;

— організація безпечного і актуального налаштування СКБД. Дана вимога включає в себе загальні завдання забезпечення безпеки, такі як своєчасна установка оновлень, відключення невикористовуваних функцій або застосування ефективної політики паролів. З'являються нові види і моделі зберігання даних.

Наступні вимоги можна назвати залежними від даних:

— безпека користувацького ПЗ. Сюди можна віднести завдання побудови безпечних інтерфейсів і механізмів доступу до даних;

— безпечна організація і робота з даними. Питання організації даних і управління ними є ключовим в системах зберігання інформації. У цю область входять завдання організації даних з контролем цілісності та інші, специфічні для СКБД проблеми безпеки. Фактично це завдання включає в себе основний обсяг залежать від даних вразливостей і захисту від них.

Для вирішення наведених проблем забезпечення інформаційної безпеки СКБД необхідно перейти від методу закриття вразливостей до комплексного підходу забезпечення безпеки сховищ інформації. Основними етапами цього переходу, повинні стати наступні положення:

— розробка комплексних методик забезпечення безпеки сховищ даних на підприємстві. Створення комплексних методик дозволить застосовувати їх при розробці та впровадженні сховищ даних і користувальницького ПО. Дотримання комплексною методикою дозволить уникнути багатьох помилок управління СКБД і захиститися від найбільш поширених на сьогоднішній день вразливостей;

— оцінка і класифікація загроз і вразливостей СКБД. Класифікація загроз і вразливостей СКБД дозволить упорядкувати їх для подальшого аналізу і захисту, дасть можливість фахівцям з безпеки встановити залежність між уразливими і

причинами їх виникнення. В результаті при введенні конкретного механізму в СКБД, у адміністраторів і розробників з’явиться можливість встановити і спрогнозувати пов’язані з ним загрози і заздалегідь підготувати відповідні засоби забезпечення безпеки;

— розробка стандартних механізмів забезпечення безпеки. Стандартизація підходів і мов роботи з даними дозволить створити засоби забезпечення безпеки, які застосовуються до різних СКБД. В даний момент вони можуть бути лише методичними чи теоретичними, так як, на жаль, поява готових комплексних програмних засобів захисту багато в чому залежить від виробників і розробників СКБД і їх бажання створювати і слідувати стандартам.

Існує цілий ряд технологій і прийомів атак на бази даних, ефективність яких залежить від конфігурації бази даних і сервера, на якому вона функціонує, від того, наскільки правильно спроектована і реалізована IT-інфраструктура і топологія мережі в цілому, від людського фактора і лояльності персоналу. Атаки на web-сервери і на сервери баз даних часто переслідують одні й ті ж цілі, запускаються одними і тими ж особами, і мають схожий характер. Тому і захист інформації в базах даних будується на використанні рішень, що мають схожі принципи роботи і архітектуру. Серед безлічі засобів захисту БД можна виділити основні і додаткові.

До основних засобів захисту інформації відносять такі:

— парольний захист;

— захист полів і записів таблиць БД;

— встановлення прав доступу до об’єктів БД;

— шифрування даних і програм.

До додаткових засобів захисту БД можна віднести такі, які не можна прямо віднести до засобів захисту, але які безпосередньо впливають на безпеку даних. Це:

— вбудовані засоби контролю значень даних відповідно до типів;

— підвищення достовірності даних, що вводяться;

— забезпечення цілісності зв’язків таблиць;

— організація спільного використання об’єктів БД в мережі.

На думку експертів компанії Application Security, існує 10 основних загроз БД, які найбільш часто ігноруються IT-персоналом:

— використання стандартних порожніх або слабких паролів і логінів;

— SQL-ін’єкції;

— розширені користувацькі і групові права;

— активізація невикористовуваних функцій БД;

— порушення в управлінні конфігураціями;

— переповнення буфера;

— ескалація привілеїв;

— DoS-атаки;

— несвоєчасне оновлення ПЗ;

— відмова від шифрування даних на стаціонарних і мобільних пристроях.

Схоже дослідження було проведено компанією NCC Group. NCC Group провела аналіз 20 звітів про огляд побудови баз даних, щоб виявити, які вразливості в безпеці найчастіше не помічаються системними адміністраторами[15].

На рисунку нижче (Рисунок 2.1) показано 10 найпопулярніших питань, які були найчастіше виявлені під час їх оцінок.

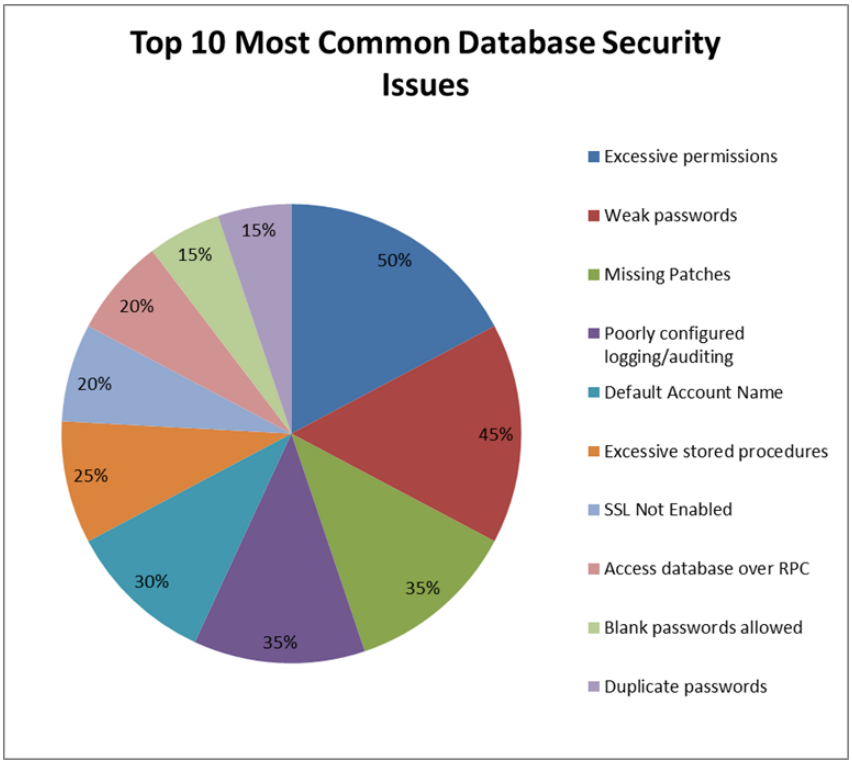

Рисунок 2.1 – 10 найбільш поширених вразливостей у БД

1. Надмірні привілеї: Було встановлено, що у 50 відсотків оцінюваних баз даних, користувачі володіли надмірними привілеями. Хоча підбір баз даних мав велику кількість облікових записів з відведеною роллю DBA, у більшості випадків було показано, що облікові записи користувачів містять привілеї за замовчуванням або їм надаються ролі, які мають доступ до функцій, які не потрібні.

2. Слабкі паролі: виявлено, що 45 відсотків баз даних мають користувачів із слабкими або типовими паролями. Слабка довіреність є проблемою для кожної організації. Системи, які не застосовують чітку політику паролів, можуть бути легко скомпрометовані. Слабка політика щодо паролів також є свідченням того, що інші системи всередині мережі можуть мати слабкі облікові дані, що розширює можлвості атаки для зловмисника. Ці паролі можуть бути легко відгадані або зламані і можуть дозволити зловмиснику отримати привілейований доступ до бази даних. Дана проблема також містилася в Топ-10 неправильних конфігурацій безпеки Windows Server на 4 місці.

3. Відсутні патчі: у 35% оцінюваних баз даних були відсутні оновлення безпеки або були запущені старі версії програмного забезпечення. Цікавим є те, що у більшості цих систем бракувало патчів, яким більше року. Це викликає питання: чи не в цьому вина власників та адміністраторів, які знаходять занадто складним застосовування відповідних патчей, або ж у відповідних організацій є невідповідна політика управління патчем. Так чи інакше, той факт, що більше як третині баз даних не вистачає цих патчів, викликає занепокоєння. Ця проблема також містилася в Топ-10 неправильних конфігурацій безпеки Windows Server на 1 місці.

4. Погано налаштований аудит: 35 відсотків баз даних були неправильно налаштовані для ведення аудиту. Це особливість, яку мають усі бази даних для відстеження та аудиту подій, таких як зміни та доступ до даних. Не відстеження подій, таких як створення облікових записів та доступ чи зміна до конфіденційних даних у виробничій системі, може ускладнити виявлення того, що сталося, якщо порушення відбулося. Хоча це питання, як правило, вважається низьким рівнем ризику, все ж важливо включати аудит при створенні бази даних. Ця проблема також містилася в Топ-10 неправильних конфігурацій безпеки Windows Server на 2 місці.

5. Назва облікового запису за замовчуванням: Тридцять відсотків баз даних містять облікові записи за замовчуванням. Як частина стратегії захисту завжди рекомендується, щоб облікові записи за замовчуванням були перейменовані та заблоковані, де це можливо, оскільки вони можуть бути використані як ціль для зламу або вгадування пароля.

6. Надмірна кількість збережених процедур: виявлено, що в 25 відсотках баз даних є надмірна кількість потенційно небезпечних збережених процедур, включаючи ті, які можуть запускати системні команди або отримати доступ до файлів на базі операційної системи. Вважається, що ця проблема становить небезпеку для безпеки бази даних, оскільки збережені процедури ефективно підвищують функціонал, який може бути використаний для запуску атак на базову операційну систему хоста і навіть на інші хости в мережі.

7. Не ввімкнений SSL: 20 відсотків баз даних приймали з'єднання через чіткі текстові канали, якщо зловмисник має доступ до мережі та здатний відслидковувати мережевий трафік, тоді може бути порушена конфіденційність та цілісність цієї передачі даних.

8. Дозволені порожні паролі: 15 відсотків баз даних були налаштовані таким чином, що дозволяли використовувати пусті паролі. Як і у випадку проблем зі слабкими паролями, жодна база даних насправді не була налаштована з порожнім паролем.

9. Дублікати паролів: у 15% баз даних були налаштовані користувачі з дублюючими паролями. Це, як правило, вказує на те, що при створенні облікового запису використовується один пароль за замовчуванням або, що користувачі не були навчені тому, як створити надійний пароль або, в гіршому випадку, як змінити свій пароль.

10. Первинний номер облікового запису є звичайним текстом: П'ять відсотків баз даних містили номера облікових записів, щ зберігалися у вигляді тексту, що є порушенням протоколу управління інформацією. Вся конфіденційна інформація повинна зберігатися в зашифрованому форматі всередині бази даних.

2.3 Сучасне апаратне та програмне забезпечення для захисту баз даних

Наразі існує безліч програмних рішень для захисту баз даних і забезпечення безпеки конфіденційної інформації:

— FortiDB;

— SafeNet ProtectDB;

— McAfee Database Security;

— Secret Disk Server NG;

— Крипто БД: захист баз даних (Oracle);

— DataSecure і інші.

FortiDB - це сучасні рішення по захисту баз даних, які допомагають великим підприємствам і хмарним сервіс провайдерам захищати свої БД і додатки від внутрішніх і зовнішніх загроз. Гнучка структура політик безпеки дозволяє FortiDB швидко і легко впроваджувати внутрішній IT контроль активності БД, проводити IT аудит і перевірку відповідності вимогам[5].

Переваги рішень FortiDB:

— масштабуюче рішення щодо захисту сотень БД розширює вбудовані можливості забезпечення безпеки БД і їх захисту від внутрішніх і зовнішніх загроз;

— підтримка різних видів платформ, включаючи AIX, Red Hat Enterprise Linux, Solaris 10, Windows XP / Vista, Windows Server 2003 і віртуальних середовищ;

— велика кількість готових політик безпеки, які покривають більшість відомих експлойтів, вразливостей в конфігураціях і операційних ризиків;

— легке налаштування і впровадження моніторингу / аудиту сотень БД за допомогою централізованої панелі управління політиками;

— збереження даних гарантується збором всіх типів активностей БД починаючи від адміністративних подій до активності користувачів незалежно від використовуваних команд або типів з'єднань;

— готові звіти допоможуть в перевірці відповідності вимогам з безпеки, таким як SOX і PCI DSS.

Модельний ряд FortiDB складається с 3 основних моделей (Рисунок 2.2):

Рисунок 2.2 – Прилади серії FortiDB

Дані прилади мають різні функціональні можливості та властивості, які представлені нижче (Рисунок 2.3).

Свойства	FortiDB – 500D	FortiDB – 1000D	FortiDB – 3000D
Количество поддерживаемых баз данных	15	30	90
Сетевые интерфейсы	4x GE RJ45 4x SFP	6x GE RJ45 2x SFP	4x GE 2x GE SFP
Объем оперативной памяти	2 GB	3 GB	4 GB
Базовый объем хранилища системы	2x 2 TB		
Максимальный объем хранилища системы	4 TB Raw, 2 TB RAID1	4 TB Raw, 2 TB RAID1	4 TB (2x 2 TB)
Поддерживаемые базы данных	DB2 UDB V8 (VA only), DB2 UDB V9.x (VA only), DB2 UDB V9.1/V9.5/V9.7 MS SQL Server 2000/2005/2008/2008R2, MS SQL Server 2012 MySQL 5.1/5.5 Oracle 9i/10gR1/10gR2/11g Sybase ASE 12.5 (sniffer only) 15.0.2/15.02/15.5/15.7 (MDA only)		
Поддерживаемые репозитории баз данных	Apache Derby 10.x, Microsoft SQL Server 2005, Microsoft SQL Server 2008, Oracle 10gR2, Oracle 11g, PostgreSQL 8.3		
Поддерживаемые браузеры	Internet Explorer 7,8,9 Firefox 3,4,5		
Горячая замена блока питания	-	+	+
Форм фактор	1U	2U	2U

Рисунок 2.3 – Властивості різних пристроїв серії FortiDB

Пристрої серії FortiDB призначені для захисту баз даних. Вони здійснюють управління уразливими, моніторинг активності баз даних (DAM), запобігання витокам даних (DLP), автоматизацію аудиту та перевірки на відповідність, контроль змін, віртуалізацію. Можливість здійснення безперервного аудиту дозволяє виявляти на ранніх стадіях не тільки «зовнішні» атаки, спрямовані на базу даних (цю функцію здійснює FortiGate), але і більш «тонкі» аномалії в її роботі - наприклад, нетипові або підозрілі транзакції, що здійснюються від імені легально зареєстрованих облікових записів.

Пристрої FortiDB доступні як в стандартному для Fortinet варіанті апаратно-програмного комплексу, так і у вигляді окремого програмного забезпечення. Серія FortiDB включає в себе три моделі: FortiDB-400B - для малих підприємств, FortiDB-1000B - для підприємств середнього розміру і FortiDB-2000B - для великих підприємств. Вони дозволяють захищати від 10 до 60 баз даних. Підтримується захист таких баз даних, як MS SQL Server (версії 2000, 2005, 2008), MySQL 5.1, Oracle (версії 9.2.x, 10gR1, 10gR2, 11.1.0.x), Sybase ASE (версії 12.0, 12.5, 15.0. 2).

Програмне рішення для захисту СКБД SafeNet ProtectDB володіє гнучкими можливостями, дозволяючи захистити дані в базах даних на рівні стовпців, на рівні додатків, і при операціях над великими масивами даних при їх трансформації або обробці[6].

Рішення інтегровано з платформою DataSecure, що дозволяє досягти продуктивності до 100 000 операцій в секунду при шифруванні баз даних, надійно захистити криптографічні ключі, знизити ризик доступу до даних з боку адміністратора СКБД і реалізувати шифрування баз даних різноманітних вендорів на основі єдиного продукту.

Рішення для шифрування баз даних SafeNet ProtectDB дозволяє захищати корпоративні і персональні дані, а також іншу інформацію, що зберігається в базах даних.

Центральний інтерфейс для налаштувань шифрування і управління ключами дозволяє шифрувати дані практично на будь-якій кількості різних баз даних, що значно підвищує рівень безпеки організації. Апаратне зберігання ключів і

шифрування підвищують захищеність і надійність систем, не знижуючи їх продуктивність.

Працюючи в зв'язці з платформою DataSecure, SafeNet ProtectDB обспечівает надійну аутентифікацію користувачів і адміністраторів, підтримуючи схему M з N і принцип поділу обов'язків.

Можливості:

— захист там, де він потрібен. Апаратне шифрування обраних стовпців в базі даних, або підмножини рядків дозволяє забезпечити надійний захист саме тих даних, які його потребують;

— централізоване управління. Адміністрування всіх ключів, політик шифрування і доступу, а також налаштувань через єдину графічну консоль значно спрощує процес управління;

— надійні алгоритми. Найвищий рівень безпеки баз даних досягається завдяки використанню перевірених криптографічних алгоритмів, докладних політик доступу і надійних механізмів аутентифікації;

— контроль доступу. Політики доступу ProtectDB дозволяють досягти відповідності законодавчим актам і стандартам, а також втілити принцип поділу адміністративних обов'язків при доступі до бази даних;

— до 100 000 операцій в секунду і більше при використанні кластера;

— єдиний простий графічний інтерфейс для управління всіма функціями;

— шифрування і захищене зберігання ключів для основних СКБД, API і операційних систем;

— інтеграція на рівнях додатки, бази даних або файлової системи;

— одночасно можуть використовуватися різні види інтеграції;

— детальна специфікація політик доступу для кожного користувача (IP-адреси, кількість кріптоопераций на годину, можливість роботи в певні дні тижня і години доби; з точністю до стовпчика при шифруванні БД);

— відповідає стандарту PCI DSS;

— сертифікований на відповідність з FIPS 140-2 Level 2 і Common Criteria Evaluation Assurance Level 2;

— комунікація між DataSecure і клієнтами опціонально відбувається по SSL / TLS з можливістю одно- або двосторонньої аутентифікації за допомогою сертифікатів і пар «логін-пароль»;

— опція авторизації особливо критичних функцій мінімум двома адміністраторами ("m of n");

— резервне копіювання, кластеризація і розподіл навантаження;

— шифрування даних за фізичними межами організації (філії, точки продажу торгових підприємств і т.д.);

— автоматизована міграція незашифрованих даних;

— періодична автоматизована ротація ключів;

— ведення докладних аудиторських та інших балок в захищеному форматі і активне оповіщення про можливі атаки;

— захист і індикація злому і фізичного доступу до ключів;

— кілька рівнів сервісної підтримки.

Переваги:

— можливості зростання. Рішення щодо шифрування баз даних ProtectDB масштабується відповідно до зростаючими потребами бізнесу, надійно захищаючи сотні мільйонів записів. Збільшення кількості клієнтів, нові партнери і бізнес можливості не будуть обмежені через технологічні недоліків;

— захист спільної роботи. Рішення дозволяє надати доступ до даних бізнес-партнерам і може працювати в зв'язки з продуктами інших вендорів, при цьому тільки аутентифіковані й авторизовані користувачі матимуть доступ до даних;

— високий рівень продуктивності. Шифрування відбувається прозоро для користувачів і бізнес-процесів, не знижуючи продуктивності бази даних;

— досвід. Досвід компанії DataSecurity Technologies по впровадженню даного рішення дозволяє провести тестування і інтегрувати рішення для шифрування баз даних в рекордно короткі терміни, без компромісів в безпеці.

Основні функції:

— управління ключами (генерація, захищене зберігання, ротація);

— захист даних в СКБД на рівні стовпців, контроль дії адміністратора;

— захист даних для додатків;

— шифрування файлів і папок на ПК і серверах;

— криптографічні операції (шифрування, ЕЦП, MAC, генерація випадкових чисел);

— можливість високошвидкісного пакетного шифрування;

— рішення інтегровано з основними СКБД, бібліотеками програмування та операційними системами, що дозволяє здійснити його впровадження за короткий час (від декількох годин до 2-3 днів).

Пітдтримувані бази даних:

— Oracle;

— Microsoft SQL Server;

— IBM DB2;

— Teradata.

Алгоритми шифрування:

— AES;

— 3DES;

— DES;

— RSA (шифрування і ЕЦП);

— RC4;

— SHA-I;

— HMACSHA-I.

Підтримувані платформи:

— Microsoft Windows;

— Linux;

— IBM z / OS;

— Sun Solaris;

— HP UX;

— IBM AIX.

McAfee Data Center Security Suite for Databases. Забезпечує захист критично важливих для комерційної діяльності баз даних в режимі реального часу від усіх видів

загроз: зовнішніх, внутрішніх і навіть від засобів використання вразливостей всередині баз даних. Програмний продукт забезпечує надійний захист і безперервне нормативно-правову відповідність без необхідності вносити зміни в архітектуру, купувати дороге апаратне забезпечення і час від часу відключати бази даних[7].

У комплект McAfee Data Center Security Suite for Databases включений ряд провідних продуктів McAfee, забезпечують комплексний захист. Можливості цих продуктів значно перевершують можливості вбудованих в бази даних функцій безпеки, що легко долаються зловмисниками. завдяки модульному характеру рішення McAfee для захисту баз даних ви можете індивідуально налаштовувати засоби захисту з метою автоматизації процесів знаходити Ваш, захистом, моніторингом і безпекою баз даних. Дане рішення не вимагає спеціальних знань систем баз даних, що дозволяє співробітникам вашого IT-підрозділу швидше досягати необхідних результатів[8].

— McAfee Vulnerability Manager for Databases автоматично виявляє всі бази даних і допомагає вам проводити оцінку потенційних вразливостей;

— McAfee Database Activity Monitoring дозволяє в режимі реального часу збирати інформацію про всі дії, пов'язані з базами даних, включаючи доступ користувачів, що мають відповідні права;

— McAfee Virtual Patching for Databases допомагає захистити бази даних від потенційних зломів ще до установки пакетів виправлень, що випускаються постачальником.

McAfee Data Center Security Suite for Databases включає в себе наступні продукти: McAfee Vulnerability Manager for Databases, McAfee Database Activity Monitoring і McAfee Virtual Patching for Databases. McAfee Data Center Security Suite for Databases дає можливість проводити повну оцінку вразливості баз даних, здійснювати їх моніторинг в режимі реального часу і забезпечувати їх захист.

McAfee Data Center Security Suite for Databases інтегрований з програмним забезпеченням McAfee ePO, що дає можливість отримувати повну картину стану системи безпеки і управляти коштами захисту без «мертвих зон». Це сама передова з наявних в галузі консолей для управління засобами захисту. Вона забезпечує повний

збір інформації про ступінь захищеності баз даних, про рівень корпоративної безпеки, про показники нормативно-правової відповідності та про порядок ведення звітності про нормативно-правовому Відповідно. Консоль McAfee ePO, що отримала широке визнання фахівців, інтегрується з іншими продуктами McAfee для управління безпекою і ризиком, а також з продуктами партнерів по McAfee Security Innovation Alliance.

Завдяки повній інтеграції з програмним забезпеченням McAfee ePO рішення для захисту баз даних більше не обмежена вузькими рамками своїх власних коштів управління. наявність єдиної консолі дозволяє легко об'єднати різні бази даних в рамках однієї програми управління.

McAfee Vulnerability Manager for Databases дає можливість повністю автоматично виявляти всі бази даних, наявні у вашому середовищі, а також виконувати їх ретельне сканування на наявність конфіденційних даних, таких як інформація про платіжні картки, паспортні дані, номери телефонів і т. д.

Також McAfee Vulnerability Manager for Databases дає можливість отримувати детальну і надійну інформацію, що дозволяє пріоритезувати і усувати прогалини в захисті, що позбавить вас від необхідності користуватися дорогими послугами сторонніх консультантів з безпеки і дозволить краще підготуватися до аудитів на відповідність нормативно-правовим вимогам.

Переваги McAfee Vulnerability Manager for Databases:

— сканування / перевірка на наявність більш ніж 4 700 видів вразливостей (в два рази більше, ніж у конкурентів), в тому числі перевірка на наявність незахищеного PL / SQL-коду і швидка перевірка паролів (в два рази швидше, ніж у конкурентів);

— перевірка на наявність тільки що виявлених вразливостей в більшості випадків не пізніше, ніж через 72 години;

— інтеграція з технологіями моніторингу баз даних (DAM), що дозволяє автоматизувати процес усунення вразливостей і процес заповнення об'єктів правил для динамічного поновлення політик.

McAfee Database Activity Monitoring дозволяє в режимі реального часу отримувати інформацію про всі дії, пов'язані з базами даних, включаючи локальний доступ користувачів з відповідними правами і витончені атаки, які виходять із середини бази даних. Продукт захищає бази даних за допомогою набору заздалегідь налаштованих засобів захисту і допомагає створити індивідуальну політику безпеки для конкретної середовища.

Переваги McAfee Database Activity Monitoring:

— глибокий аналіз пам'яті, що зберігає план виконання операторів, який дозволяє максимізувати видимість всіх джерел атак і захист від них;

— відстеження зовнішніх загроз, загроз з боку привілейованих користувачів і витончених атак зсередини баз даних;

— зниження рівня ризику і відповідальності за порушення безпеки завдяки припиненню атаки до того, як вона може завдати шкоди;

— економія часу і грошей завдяки прискореному розгортання і більш ефективної архітектурі;

— неінтрузівна архітектура, для установки, оновлення та видалення якої не потрібно відключати бази даних.

McAfee Virtual Patching for Databases. McAfee Virtual Patching for Databases захищає бази даних від ризику, пов'язаного з наявністю невиправлених вразливостей. McAfee Virtual Patching for Databases захищає бази даних с невстановленими пакетами виправлень шляхом виявлення та блокування спроб атак і вторгнень в режимі реального часу, не вимагаючи відключення баз даних і тестування додатків. За допомогою даного рішення для установки віртуальних виправлень ви можете забезпечити безперервну захист баз даних, використовують застарілі версії СКБД, вже не підтримувані постачальниками, і тим самим продовжити термін життя застарілих баз даних і заощадити час і гроші організації.

Переваги McAfee Virtual Patching for Databases:

— інтеграція технології DAM, що забезпечуєповний захист від атак, проведених по будь-яким векторах з використанням будь-яких передових методів;

— захист від відомих вразливостей і засобів використання вразливостей без необхідності вносити будь-які зміни в бази даних і додатки;

— забезпечення захисту McAfee через 48-72 години після публікації уразливості або випуску пакета виправлень виробником бази даних.

Secret Disk Server NG – комплекс захисту конфіденційної інформації та корпоративних баз даних на серверах від несанкціонованого доступу, копіювання, ушкодження, крадіжки або неправомірного вилучення. Система Secret Disk Server NG надійно захищає дані і приховує сам факт їх наявності на сервері. При запису даних на диск відбувається їх зашифрование, при читанні – розшифрування. Знаходяться на диску дані завжди зашифровані, що робить доступ до них неможливим для зловмисника, навіть якщо він отримає фізичний доступ до сервера або жорсткого диска.

Secret Disk Server NG може бути використаний як самостійне рішення, а також як елемент комплексної системи захисту конфіденційної інформації для вирішення наступних завдань:

— розмежування доступу до даних, що зберігаються і обробляються на серверах додатків, наприклад, файлів баз даних, поштових сховищ і ін. Управління доступом до даних дозволяє гнучко розмежувати роботу співробітників організації з захищеною інформацією;

— приховування конфіденційної інформації на сервері забезпечить додатковий рівень захисту, так як зашифровані розділи дисків виглядають поза Secret Disk Server NG як розділеного простору.

Secret Disk Server NG необхідний:

— для захисту серверів в разі централізованого зберігання і обробки цінної корпоративної інформації від таких загроз як:несанкціонований доступ до конфіденційних даних на захищається сервері або сховище даних; копіювання даних нелояльних або підкупленим співробітником, який може мати фізичний доступ до сервера; несанкціоноване копіювання даних по мережі підприємства IT-фахівцем, що має розширені права адміністратора;

— для екстреного блокування доступу до конфіденційної інформації в надзвичайних ситуаціях в разі фізичного захоплення сервера;

— для реалізації захисту високонавантажених, відмовостійких рішень. Secret Disk Server NG підтримує технологію багатопотокового шифрування, максимально оптимізуючи обчислювальні ресурси сучасних багатопроцесорних систем. Продукт дозволяє створювати резервні копії зашифрованих розділів і відкритих в ексклюзивному режимі файлів без зупинки працюють сервісів і додатків (Microsoft Exchange, Microsoft SQL Server і ін.).

Ключові переваги Secret Disk Server NG

Дані на носіях залишаються завжди зашифрованими, що робить доступ до них неможливим, навіть якщо зловмисник отримає фізичний доступ до сервера або жорстких дисків.

— використовується вбудована в ядро ОС Windows криптографія або швидкі і надійні криптоалгоритми з пакету розширення Secret Disk Crypto Extension Pack;

— доступна установка додаткових пакетів від сторонніх постачальників криптографії (криптопровайдерів), включаючи КріптоПро CSP, Signal-COM CSP, Infotecs CSP, які реалізують російські стандарти алгоритмів шифрування;

— використовується метод двофакторної аутентифікації адміністратора Secret Disk Server NG за допомогою електронного ключа і пароля;

— екстрене блокування доступу до даних по сигналу "тривога". Сигнал може бути поданий як зовнішнім пристроєм (наприклад, "червоною кнопкою", радіо-брелоком, охоронною сигналізацією або по GSM-каналу *), так і з клавіатури комп'ютера або мишею. Реакцію на сигнал "тривога" можна налаштувати як для сервера в цілому, так і для кожного зашифрованого диска окремо. Перед відключенням диска за сигналом "тривога" може бути виконана зупинка сервісів і служб (наприклад, MS SQL Server, MS Exchange Server і т.п.);

— є можливість заборони мережевого доступу до захищається даними, що запобігає копіювання важливих даних зловмисником з розширеними правами адміністратора мережі.

Надійність:

— рішення стійке до можливих збоїв операційної системи або відключень електроживлення, що виключає пошкодження даних;

— підтримуються відмовостійкі кластерні конфігурації сервера, що захищається;

— підтримується резервне копіювання і відновлення ключів шифрування на випадок втрати електронного ключа;

— передбачено захист від випадкового форматування відключеного зашифрованого диска.

Зручність. Гнучке і зручне адміністрування дозволяє спростити роботу адміністраторів:

— підтримка широкого спектру накопичувачів дозволяє захищати окремі жорсткі диски сервера, будь-які дискові масиви (SAN, програмні та апаратні RAID-масиви), а також знімні диски;

— рішення допускає необмежену кількість зареєстрованих адміністраторів;

— підтримується управління через консоль Microsoft Windows (MMC) або віддалений робочий стіл (RDP);

— підтримуються індивідуальні сценарії для кожного зашифрованого диска. Індивідуальні сценарії можуть виконуватися перед підключенням диска, після підключення, перед відключенням, після відключення, що дозволить домогтися необхідної реакції від системи;

— використання методу прозорого шифрування інформації дозволяє не зупиняти роботу сервера під час первісного зашифрування даних.

Підтримувані платформи (Робоча станція адміністратора):

— Microsoft Windows Server 2016;

— Microsoft Windows Server 2012 R2;

— Microsoft Windows Server 2012;

— Microsoft Windows Server 2008 R2;

— Microsoft Windows Server 2008;

— Microsoft Windows 8.1;

— Microsoft Windows 8;

— Microsoft Windows 7;

— Microsoft Windows Vista.

Типи підтримуваних дисків:

— основні розділи та логічні диски жорстких дисків;

— томи динамічних дисків;

— зйомні диски (USB-диски, магнітооптика і ін.);

— віртуальні диски (файли-контейнери);

— зовнішні сховища (SAN).

Типи файлових систем:

— NTFS;

— FAT 32;

— FAT 16.

2.4 Вбудовані засоби безпеки в нереляційних базах даних на прикладі MongoDB

MongoDB – не реляційна база даних, відмінна від бази даних на основі мови запитів SQL. Це означає, що користувачі можуть вводити дані в MongoDB, не визначаючи таблиці і поля і не призначаючи індекси. У сховищ даних такого типу багато переваг, в тому числі можливість додавати інформацію про одиничної записи, яка не має відповідного стовпчика.

Бази даних, відмінні від SQL, є добрим репозиторієм для Великих Даних, оскільки вони проектувалися для зберігання величезних обсягах не реляційних даних і швидко масштабуються відповідно до різними потребами компанії. Жоден тип бази даних не кращий за інший, вони просто призначені для різних завдань, однак при шифруванні корисно мати уявлення про їх схожість і відмінності.

При виборі методу шифрування користувач може шифрувати весь файл бази даних, окремі стовпці або дані на рівні додатку, перш ніж ввести дані в базу даних.

Найнадійніший метод – шифрування даних на рівні додатку, проте іноді дуже важко, якщо взагалі можливо, вбудувати шифрування в сторонній додаток.

Найпоширеніший метод шифрування баз даних - на рівні файлів. Таким чином, базу даних можна непомітно зашифрувати в сховище, а адміністратору баз даних зручно обслуговувати резервні копії.

Баз даних, не збудованим на базі SQL, таким як MongoDB, почасти властиві ті ж проблеми. Головна відмінність між двома типами в тому, що, оскільки в НЕ реляційних базах даних дані не розділені за стовпцями, шифрування на рівні стовпців неможливо. Тому користувачі можуть шифрувати дані тільки на рівні додатку або бази даних.

І знову, так як організувати шифрування на рівні додатку часто буває складно і дорого, переважно шифрувати всю базу даних на підсистемі сховища. На щастя для користувачів, MongoDB у своєму розпорядженні власні функціями шифрування, тому за захист конфіденційних даних додатково платити нс потрібно. Продукт пройшов вичерпне тестування і доповнений засобами оптимізації продуктивності.

MongoDB розпорядженні рішеннями для шифрування як при пересиланні, так і в сховище.

Для захисту даних, що пересилаються всі версії MongoDB підтримують протоколи TLS (Transport Layer Security) і SSL (Secure Socket Layer) для прийому і передачі даних по мережі. TLS і SSL - варіанти шифрування, широко використовувані для захисту трафіку веб-сайтів і обміну файлами.

Це протоколи шифрування для захисту даних, що пересилаються з однієї точки в іншу; однак, перш ніж дані відправляються, і після того, як прибувають в кінцеву точку, вони виявляються не зашифрованими. MongoDB надає велику документацію по налаштуванню протоколів TLS і SSL з використанням сертифікатів і пари відкритого і закритого ключів, так званою системою асиметричних ключів.

Для шифрування неактивних даних в MongoDB Enterprise використовується власний механізм шифрування з симетричним ключем на основі сховища на рівні файлів. Шифрування всієї бази даних також називається прозорим шифруванням даних (TDE). Починаючи з версії 3.2 MongoDB використовує 256-розрядний

алгоритм шифрування AES з єдиним ключем, застосовуваним для шифрування і відновлення даних[11].

При шифруванні даних з використанням TDE важливо знати, як ключі шифрування зберігаються в MongoDB. Коли адміністратор шифрує файл бази даних, формується унікальний закритий ключ шифрування. Для кожного зашифрованого файлу бази даних створюється новий закритий симетричний ключ, і всі ключі на пристрої зберігання даних шифруються з використанням головного ключа.

Ключі бази даних зберігаються разом з зашифрованими даних, але MongoDB ніколи не дозволяє зберігати головний ключ на одному сервері з зашифрованими даними. Це означає, що адміністратор бази даних або безпеки повинен ідентифікувати безпечне місце зберігання для головного ключа шифрування.

При роботі з максимальним навантаженням середня затримка підсистеми зберігання з шифруванням становить від 10 до 20%, в залежності від кількості даних, що читаються або записуються користувачем в базу даних. Коли користувач записує в базу даних тільки великі обсяги даних, вплив на продуктивність велике; однак набагато частіше користувач виконує в основному команди тільки для читання даних, і в більшості організацій цей показник складе, швидше за все, 5-10%.

З метою оптимізації шифрування MongoDB шифрує кожну базу даних з використанням зашифрованою підсистеми зберігання WiredTiger. MongoDB придбала WiredTiger в 2014 році, і вона стала підсистемою зберігання за замовчуванням для MongoDB починаючи з версії 3.2.

Коли користувач читає або записує дані в зашифровану базу даних, дія зачіпає тільки сторінку, на якій збережено дані, а не всю базу даних. Крім того, знижується вплив на продуктивність завдяки обмеженню кількості даних, які доводиться шифрувати і розшифровувати, щоб зашифрувати і розшифрувати один фрагмент даних.

Таким чином, MongoDB надає потужне рішення для шифрування неактивних даних, яке б задовольнило потреби безпеки і продуктивності більшості користувачів. Тестування NIST FIPS забезпечує відповідність вимогам нормативних актів, а

передова підсистема зберігання WiredTiger автоматично враховує мінливі потреби користувачів в сфері безпеки.

Проте у NoSQL базах даних відсутні функціонал, що забезпечує конфіденційність і цілісність даних. Дані в базах такого типу є неструктурованими, отже, доступ до полів і рядках матимуть абсолютно всі користувачі. Це може привести до дублювання даних і ускладнити підтримку послідовності даних.

Транзакції в NoSQL базах записуються не відразу. Виникає можливість перетину транзакцій.

До типових вразливостей NoSQL баз даних відносять:

— переповнення буфера;

— перевищення привілеїв;

— зберігання інформації в незашифрованому вигляді;

— лазівки системи ідентифікації / аутентифікації;

— API. Доступ до NoSQL-СКБД за допомогою бібліотек. Часто бібліотеки мають відкритий вихідний код.

Оскільки існує більше 20 різних реалізацій NoSQL, відсутність стандартів також підвищує складність підтримки безпеки даних. Конфіденційність і цілісність даних повинні повністю забезпечуватися додатком, яке звертається до даних NoSQL. Це погано, коли остання лінія захисту будь-яких цінних даних знаходиться на рівні додатку. Розробники додатків не відрізняються уважністю до реалізації функцій безпеки, і новий програмний код зазвичай означає нові помилки. Будь-які запити, що направляються в базу даних NoSQL, повинні перенаправлятися, фільтруватися і підтверджуватися, в той час як сама БД повинна завжди знаходиться в захищеному середовищі.

Якщо ключовими вимогами організації до БД є масштабованість і доступність, то система NoSQL може бути правильним вибором для певних датасетів. Однак, архітекторам систем слід ретельно розглядати свої вимоги до безпеки, конфіденційності та цілісності перед вибором бази даних NoSQL. Відсутність в NoSQL функцій безпеки, а саме, підтримка аутентифікації або авторизації, означає, що конфіденційні дані найкраще зберігати в стандартних RDBMS.

2.5 Вбудовані засоби безпеки в реляційних базах даних

2.5.1 Засоби безпеки в MSSQL

SQL Server підтримує два режими аутентифікації: за допомогою Windows і за допомогою SQL Server. Перший режим дозволяє реалізувати рішення, засноване на одноразовій реєстрації користувача і єдиному паролі при доступі до різних програм (Single SignOn solution, SSO). Подібне рішення спрощує роботу користувачів, позбавляючи їх від необхідності запам'ятовування безлічі паролів і тим самим знижуючи ризик їх небезпечного зберігання (згадаємо стікери з паролями, наклеєні на монітори). Крім того, даний режим дозволяє використовувати засоби безпеки, що надаються операційною системою, такі як застосування групових і доменних політик безпеки, правил формування і зміни паролів, блокування облікових записів, застосування захищених протоколів аутентифікації за допомогою шифрування паролів (Kerberos або NTLM).

Аутентифікація за допомогою SQL Server призначена головним чином для клієнтських додатків, що функціонують на платформах, відмінних від Windows. Цей спосіб вважається менш безпечним, але в SQL Server він підтримує шифрування всіх повідомлень, якими обмінюються клієнт і сервер, в тому числі за допомогою сертифікатів, згенерованих сервером. Шифрування також підвищує надійність цього способу аутентифікації. Для облікового запису SQL Server можна вказати такий параметр, як необхідність змінити пароль при першому з'єднанні з сервером[12].

Не перший десяток років принцип розподілу прав доступу до об'єктів баз даних в більшості серверних СКБД заснований на наявності у кожного об'єкта бази даних користувача-власника, який може надавати іншим користувачам права доступу до об'єктів бази даних. При цьому набір об'єктів, що належать одному і тому ж користувачеві, називається схемою. Даний спосіб володіння об'єктами створював певні незручності при супроводі додатків, що використовують бази даних. Так, при звільненні працівника, який створив об'єкти, що використовуються багатьма

користувачами, і видаленні відповідної облікового запису доводилося вносити зміни в серверний код (а нерідко і в код клієнтського додатка). Розуміння можливості виникнення цих проблем призвело до поширення небезпечних, але простих у застосуванні способів управління обліковими записами користувачів. Аж до зберігання їх імен і паролів в звичайних таблицях, що різко підвищувало загрозу несанкціонованого доступу до даних і додатків[10].

У SQL Server концепція ролей розширена: ця СКБД дозволяє повністю відокремити користувача від схем і об'єктів бази даних. Тепер об'єкти бази даних належать не користувачеві, а схемі, яка не має ніякого відношення ні до яких облікових записів і тим більше до адміністративних привілеїв. Таким чином, схема стає механізмом угруповання об'єктів, що спрощує надання користувачам прав на доступ до об'єктів.

Для спрощення управління правами доступу в більшості серверних СКБД застосовується механізм ролей – наборів прав доступу до об'єктів бази даних, що привласнюються деякої сукупності користувачів. При використанні ролей управління розподілом прав доступу до об'єктів між користувачами, які виконують однакові функції і застосовують одні й ті ж додатка, істотно спрощується: створення ролі і одноразове призначення їй відповідних прав здійснюється набагато швидше, ніж визначення прав доступу кожного користувача до кожного об'єкту. SQL Server 2005 дозволяє створювати так звані вкладені ролі, тобто привласнювати однієї ролі іншу з усіма її правами. Це спрощує управління не тільки правами користувачів, а й самими ролями, створюючи, наприклад, подібні між собою групи ролей.

SQL Server також підтримує так звані ролі для додатків (application roles), які можуть використовуватися для обмеження доступу до об'єктів бази даних в тих випадках, коли користувачі звертаються до даних за допомогою певних програм. На відміну від звичайних ролей, ролі для додатків, як правило, неактивні і не можуть бути присвоєні користувачам. Їх застосування виявляється зручним в тому випадку, коли вимоги безпеки єдині для всіх користувачів, при цьому не потрібно аудит чи інша реєстрація діяльності конкретних користувачів в базі даних.

SQL Server містить вбудовані засоби шифрування, цифрового підпису та верифікації даних за допомогою симетричних ключів (алгоритми шифрування RC4, RC2, DES, AES) і асиметричних ключів (алгоритм RSA). Весь трафік між клієнтом і сервером за замовчуванням шифрується із застосуванням протоколів IP Security (IP SEC) і Secure Sockets Layer (SSL), причому подібна функціональність доступна у всіх редакціях продукту. SQL Server 2005 дозволяє при необхідності визначити політику безпеки, повністю забороняє обмін незашифрованими даними між клієнтом і сервером, що знижує ризик витоку даних, отриманих шляхом перехоплення трафіку.

Протокол SSL підтримується за допомогою інтеграції зі службами Internet Information Services (IIS) або за допомогою сервера сертифікатів, що підтримує стандарт X. 509v3 і входить до складу SQL Server. Згенеровані сертифікати не тільки використовуються для створення SSL-з'єднань – їх застосовує і SQL Service Broker.

SQL Server 2005 дозволяє здійснити захист даних на рівні колонок за рахунок шифрування інформації, що зберігається в них інформації. Він підтримує також шифрування самих даних, що зберігаються, повністю інтегроване з інфраструктурою управління ключами. Для цієї мети служать вбудовані функції EncryptByCert, DecryptByCert, EncryptByKey, DecryptByKey, EncryptByAssym, DecryptByAssym, що дозволяють використовувати шифрування за допомогою сертифіката, симетричного і асиметричного ключів в коді Transact-SQL. Необхідно, проте, пам'ятати про те, що шифрування даних може привести до втрати продуктивності, тому при створенні рішень рекомендується шифрувати тільки конфіденційні дані і здійснювати тестування продуктивності готового рішення[19].

Незважаючи на те, що SQL Server задовольняє практично всім сучасним вимогам забезпечення безпеки, саме по собі впровадження цього продукту не захистить компанію від загроз. Питання безпеки при використанні СКБД не є чисто технологічними – проблеми з їх рішенням часто виникають у зв'язку недостатню компетентність, а то і недобросовісності людей, які застосовують СКБД, що створюють рішення на їх основі або впроваджують ці рішення, так само як і изза дій співробітників-інсайдерів, свідомо порушують правила безпеки з метою отримання особистої вигоди. Крім того, забезпечення «жорсткого» режиму безпеки часто

ускладнює виконання співробітниками їх завдань, ускладнюючи доступ до необхідних для цього даних і функцій.

SQL Server не єдина хороша серверна СКБД на ринку програмного забезпечення. Надійні і прості в застосуванні СКБД масштабу підприємства випускають такі компанії, як IBM, Oracle, Sybase. Засоби забезпечення безпеки в SQL Server 2005 також не унікальні - всі перераховані вище виробники СКБД піклуються про безпеку своїх продуктів не менше, ніж корпорація Microsoft. Дане твердження ілюструється наведеної вище таблицею, в якій представлено наявність засобів забезпечення безпеки в SQL Server 2005 і в різних редакціях Oracle 10g. Порівняльна таблиця зазначених СКБД предсталена нижче (Рисунок 2.4).

Механизмы безопасности	SQL Server 2005 (все редакции)	Oracle 10g Standard Edition	Oracle 10g Enterprise Edition	Oracle 10g Enterprise with Advanced Security Option
Аутентификация с помощью Windows	✓			✓
Шифрование трафика	✓			✓
Шифрование данных	✓			✓
Поддержка открытых ключей	✓			✓
Поддержка Kerberos	✓			✓
Наличие схем, не связанных с учетными записями	✓	✓	✓	✓
Роли	✓	✓	✓	✓
Аудит	✓	✓	✓	✓
Политики	✓	✓	✓	✓
Службы сертификатов	✓	✓		
Выполнение кода с минимальными привилегиями	✓	✓	✓	✓

Рисунок 2.4 – Порівняльна таблиця механізмів безпеки СКБД

2.5.2 Засоби безпеки в MySQL

В MySQL користувачів можна створити двома способами: За допомогою операторів призначених для створення користувачів, таких як CREATE USER або GRANT. Ці оператори дозволяють вносити в таблицю привілеїв відповідні зміни. І

прямими маніпуляціями в MySQL таблицях привілеїв за допомогою операторів INSERT, UPDATE, або DELETE[14].

Кращим методом є використання операторів створення користувачів, тому що вони більш короткі і менш схильні до помилок, ніж прямі маніпуляції в таблицях привілеїв.

Також є можливість додавати користувачів за допомогою програм сторонніх виробників, що дозволяють виробляти адміністрування в MySQL. Однією з таких програм є phpMyAdmin.

Команди GRANT і REVOKE дозволяють системним адміністраторам створювати користувачів MySQL, а також надавати права користувачам або позбавляти їх прав на чотирьох рівнях привілеїв:

Глобальний рівень. Глобальні привілеї застосовуються до всіх баз даних на зазначеному сервері. Ці привілеї зберігаються в таблиці mysql.user.

Рівень бази даних. Привілеї бази даних застосовуються до всіх таблиць зазначеної бази даних. Ці привілеї зберігаються в таблицях mysql.db і mysql.host.

Рівень таблиці. Привілеї таблиці застосовуються до всіх стовпцях зазначеної таблиці. Ці привілеї зберігаються в таблиці mysql.tables_priv.

Рівень стовпчика. Привілеї стовпчика застосовуються до окремих стовпців зазначеної таблиці. Ці привілеї зберігаються в таблиці mysql.columns_priv.

Привілеї для таблиці або стовпця формуються за допомогою логічного оператора OR з привілеїв кожного з чотирьох рівнів. Наприклад, якщо в таблиці mysql.user зазначено, що у користувача є глобальний привілей, цей привілей не відміняється на рівні бази даних, таблиці або стовпця.

Привілеї для стовпця можуть бути обчислені в такий спосіб:

— глобальні привілеї;

— OR (привілеї бази даних AND привілеї віддаленого комп'ютера);

— OR привілеї таблиці;

— OR привілеї стовпчика.

Можна зберігати криптовані дані в базі MySQL, здійснюючи шифрування і дешифрування за допомогою вбудованих інструментів. Найкраще для цієї мети

підходять функції AES_ENCRYPT () і AES_DECRYPT (), які перетворять рядки в зашифровані двійкові послідовності і назад. Ці функції реалізують симетричне криптування: для шифрування і дешифрування застосовується один і той же ключ.

Копіювання файлів БД MySQL. Базу даних MySQL можна скопіювати, якщо тимчасово вимкнути MySQL-сервер і скопіювати файли з папки / var / lib / mysql / db /. Якщо сервер не вимкнути, – імовірна втрата і псування даних. Для великих навантажених БД ймовірність втрати даних близька до 100%. Створення резервної копії БД – бекапа (англ. Backup copy) великого обсягу може зайняти тривалий час. У багатозадачних або багатокористувацьких системах, під час резервного копіювання може відбуватися запис або зміна файлів і директорій, що може привести до невірної резервної копії даних. Одним з методів безпечного створення бекапа є заборона запису в дані, які підлягають резервному копіюванню, на час створення резервної копії. Ще одним з методів є зупинка всіх додатків, які можуть змінювати дані БД, або блокування цих додатків форсованим включенням режиму тільки читання засобами інтерфейсу програмування додатків (API) ОС.

2.6 Висновки до розділу 2

В даному розділі було розглянуто та проаналізовано проблеми, які демонструють необхідність проектування системи підтримки прийняття рішень при анлізі інформаційного забезпечення безпеки даних енергетичних процесів та систем.

У цьому розділі було зрблено огляд сучасних систем, виявлені переваги і недоліки, розглянуто їх принципи роботи та ситуації, де їх варто використовувати. Окрім цього, зроблена їх порівняльна характеристика. Можна, засвідчити, що ці системи є, в осномному, імпортними, і, як наслідок, доволі дороими. Вітчизняних аналогів не було виявлено.

Як висновок з другого розділу після досліджених матеріалів було описано та обґрунтовано необхідність та важливість проектування та розробки подібної системи підтримки прийняття рішень, яка буде водночас відносно простою та ефективною.

3. МЕТОДИ РЕАЛІЗАЦІЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПРИ АНАЛІЗІ ПРОБЛЕМИ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ДАНИХ

3.1 Засоби розробки

3.1.1 Середовище розробки Visual Studio

Visual Studio IDE – це повнофункціональна платформа розробки для багатьох операційних систем, а також для Інтернету та хмари. Вона дозволяє користувачам легко користуватися інтерфейсом та можливостями середовища, щоб вони могли швидко та точно писати код.

У комплект входять наступні основні компоненти:

— Visual Basic.NET – для розробки додатків на VisualBasic;

— Visual C ++ – на традиційній мові C ++;

— Visual C # – на мові C # (Microsoft);

— Visual F# – на F# (Microsoft Developer Division).

Функціональна структура середовища включає в себе:

— редактор вихідного коду, який включає в себе безліч додаткових функцій, як автодоповнення IntelliSense, рефракторинг коду тощо;

— налагоджувач коду;

— редактор форм, передбачений для спрощеного конструювання графічних інтерфейсів;

— веб-редактор;

— дизайнер класів;

— дизайнер схем баз даних.

Visual Studio також дозволяє створювати та підключати сторонні доповнення (плагіни) для розширення функціональних можливостей практично на кожному рівні, включаючи додавання систем контролю версій (Subversion і VisualSourceSafe), додавання нових наборів інструментів (для редагування та візуального проектування коду на предметно-орієнтованих мовах програмування або інструментів для інших аспектів процесу розробки програмного забезпечення).

За допомогою Visual Studio IDE розробники також мають доступ до безлічі інструментів налагодження. Вони допомагають їм в аналізуванні помилок та їх швидкій та ефективній діагностиці. Таким чином вони можуть впевнено розгортати свої програми, знаючи, що вони усунули все, що може призвести до помилок у роботі.

Більше того, Visual Studio IDE також є тестовою платформою. Тут розробники можуть імітувати, як їх програми працюватимуть у своїх цільових середовищах, і гарантувати, що вони роблять це безперебійно після їх розгортання.

Середовище пропонує інтерфейс з багатьма можливостями, який при цьому інтуїтивно зрозумілий для користувача (Рисунок 4.1).

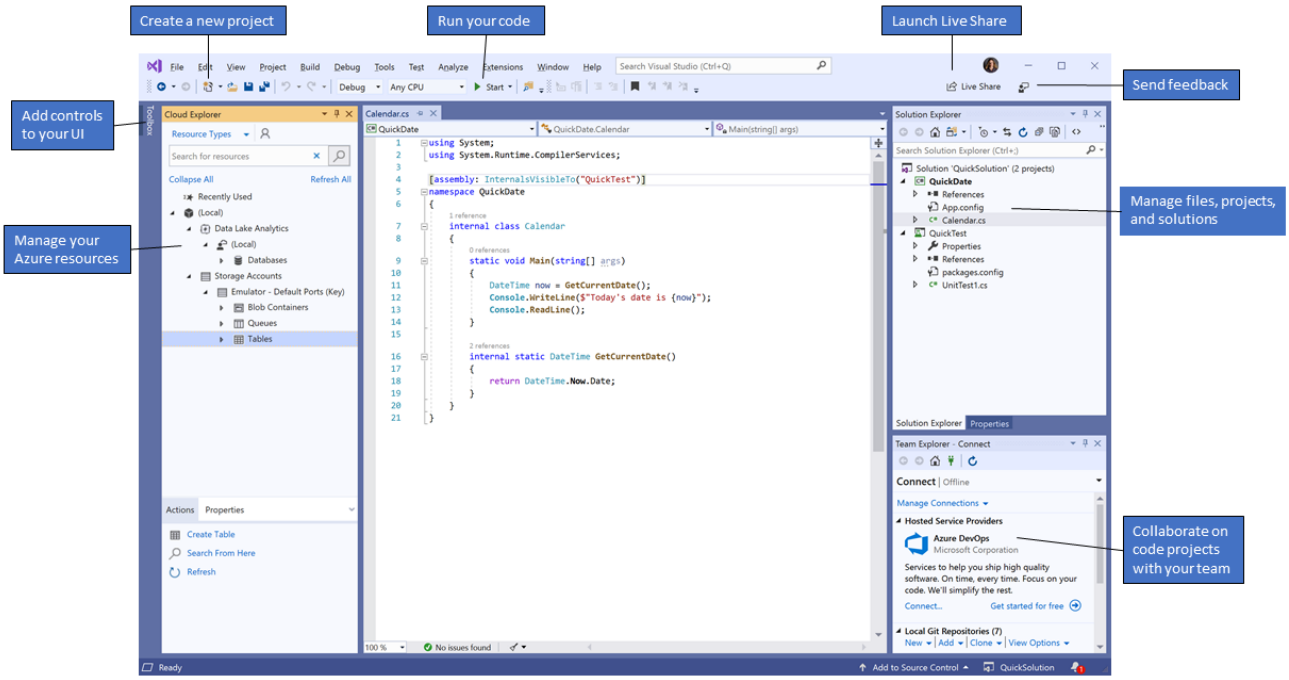

Рисунок 3.1 – Середовище розробки Visual Studio

Середовище розробки Visual Studio має безліч переваг.

Точне написання коду. За допомогою Visual Studio IDE користувачам надається допомога з написанням коду в реальному часі незалежно від мови програмування, яку вони використовують. Вбудована IntelliSense платформа пропонує підказки та описи API та автоматично доповнює рядки для більшої ефективності в написанні коду.

Крім того, Visual Studio IDE зберігає місце останньої модифікації коду розробником, якщо той досліджує решту свого коду.

Швидке налагодження. Пошук та діагностика помилок може бути проблемою, однак у Visual Studio IDE є безліч інструментів, які полегшують її. Платформа підтримує налагодження для всіх включених мов. Процес також може здійснюватися як локально, віддалено, так і в середині розробки. Це дозволяє розробникам розгортати програми на робочому столі або емулятори на мобільних пристроях та інші методи налагодження.

Можливості тестування. Visual Studio IDE оснащений платформою для тестування додатків, яка дозволяє розробникам переконатися, що вони готові розгорнути високоякісні продукти. Вони можуть це робити на бажаній мові або фреймворку.

Командна робота. Розробники Visual Studio IDE розуміють, що зазвичай є необхідність для розробки в команді. Ось чому платформа має спільні можливості, що підвищують продуктивність команди. Ці інструменти тісно інтегровані з життєвим циклом розробки проектів.

Крім того, Visual Studio IDE добре працює в режимі спільної роботи незалежно від бажаної платформи кожного учасника.

Параметри налаштування. Visual Studio IDE пропонує налаштування для кожного користувача. Вони можуть розширити функціональні можливості платформи за допомогою розширень та доповнень, доступних у Visual Studio Marketplace. Розробники навіть можуть публікувати власні розширення.

Visual Studio дозволяє отримувати доступ до документації MSDN прямо з середовища IDE. У разі, наприклад, виникнення сумнівів з приводу призначення того

чи іншого ключового слова під час роботи з текстовим редактором, можна виділити це ключове слово і натиснути клавішу <F1>, в результаті чого Visual Studio автоматично підключиться до MSDN і відобразить відповідні розділи довідки. Аналогічно, якщо потрібно подивитися, що означає та чи інша помилка компіляції, потрібно виділяти повідомлення з помилкою і натиснути <F1>.

Також Visual Studio містить графічні редактори і конструктори XML, забезпечує підтримку розробки програм Windows, орієнтованих на мобільні пристрої, підтримку розробки програм Microsoft Office і Windows Workflow Foundation, містить вбудовану підтримку рефакторинга коду і інструменти візуального конструювання класів.

3.1.2 Графічний інструмент для роботи з базами даних MySQL Workbench

MySQL Workbench – це графічний інструмент для роботи з серверами та базами даних MySQL(Рисунок 4.2). MySQL Workbench повністю підтримує версії сервера MySQL 5.6 та новіші. Він також сумісний із старими версіями сервера 5.x MySQL, за винятком певних ситуацій (наприклад, відображення списку процесів) через змінені системні таблиці. Він не підтримує версії сервера MySQL 4.x.

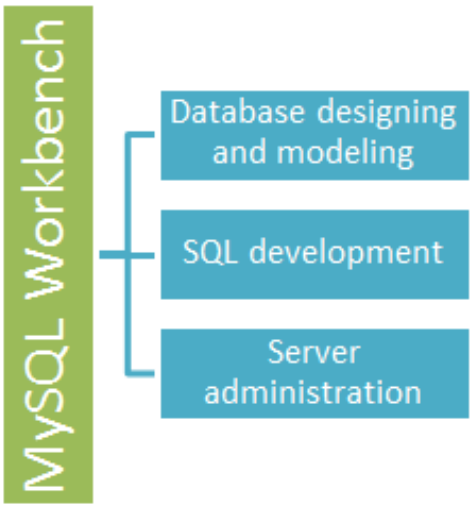

Рисунок 3.2 – Структура MySQL Workbench

Функціонал MySQL Workbench охоплює п’ять основних тем:

— розробка SQL: дозволяє створювати та керувати з'єднаннями з серверами баз даних. Поряд з можливістю налаштування параметрів з'єднання, MySQL Workbench надає можливість виконувати запити SQL на підключеннях до бази даних за допомогою вбудованого редактора SQL;

— моделювання даних (дизайн): дозволяє створювати моделі схем баз даних графічно, редагувати всі аспекти баз даних за допомогою повноцінного редактора таблиць. Редактор таблиць надає прості у використанні засоби для редагування таблиць, стовпців, покажчиків, тригерів, розділення, параметрів, вкладок та привілеїв, рутинних програм та представлень;

— адміністрування сервера: дає змогу керувати екземплярами сервера MySQL шляхом адміністрування користувачів, виконання резервного копіювання та відновлення, перевірки даних аудиту, перегляду стану баз даних та контролю за роботою сервера MySQL;

— міграція даних: Дозволяє переходити з Microsoft SQL Server, Microsoft Access, Sybase ASE, SQLite, SQL Anywhere, PostreSQL та інших RDBMS таблиць, об'єктів і даних до MySQL. Міграція також підтримує перехід від попередніх версій MySQL до останніх версій.

MySQLworkbench має зрозумілий інтерфейс та інструменти, які дозволяють розробникам та адміністраторам баз даних візуально створювати фізичні моделі дизайну баз даних (Рисунок 4.3), які можна легко перевести в бази даних MySQL. MySQL Workbench підтримує створення декількох моделей в одному середовищі.

Він підтримує всі об'єкти, такі як таблиці, представлення даних, збережені процедури, тригери тощо, які складають базу даних.

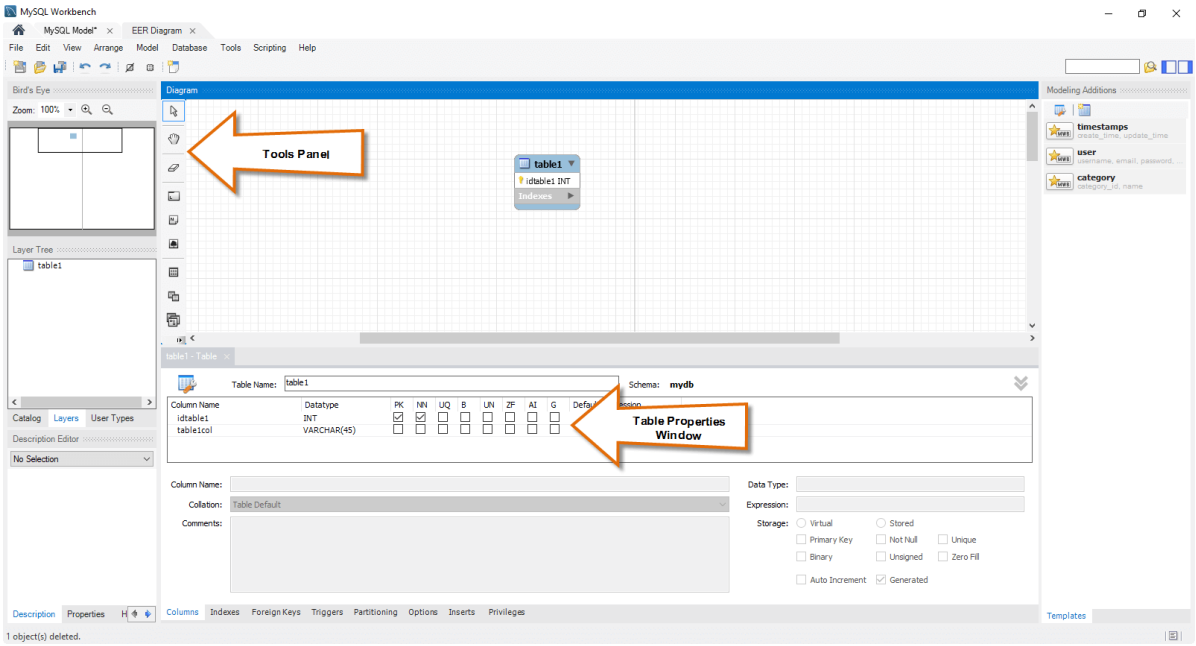

Рисунок 3.3 – Вікно моделювання в MySQL Workbench

3.1.3 Система керування базами даних MySQL

MySQL і SQL не є однаковими. SQL в MySQL розшифровується як структурована мова запитів. Це стандартна мова, яка використовується для взаємодії з базою даних. MySQL – це система управління реляційними базами даних, яка допомагає маніпулювати базою даних, що зберігаються в різних таблицях на комп'ютері.

Система керування базами даних MySQL – реляційна система керування базами даних (СКБД). Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (PostgreSQL) [8]. Порівняно з іншими проектами з відкритим кодом, такими як Apache або FreeBSD, MySQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення. Сервер MySQL написаний на мові програмування C. Зазвичай розповсюджується у вигляді набору текстових файлів із сирцевим кодом. Для інсталяції необхідно відкомпілювати файли на своєму комп’ютері і скопіювати в деякий каталог. Весь процес детально описаний в

документації. В MySQL є підтримка індексів наступних типів: B-дерево, хеш, R-дерево, GiST, GIN. При необхідності можна створити нові типи індексів [9].

MySQL підтримує великий набір вбудованих типів даних:

— числові типи;

— цілі;

— з фіксованою крапкою;

— з нефіксованою крапкою;

— грошовий тип;

— символьні типи довільної довжини;

— двійкові типи (включаючи BLOB);

— типи дата/час;

— булевий тип;

— перерахування;

— геометричні примітиви;

— мережеві типи;

— IP і IPv6-адреси;

— CIDR-формат;

— MAC-адреса;

— UUID-ідентификатор;

— XML-дані;

— JSON-дані;

— масиви;

— OID-типи;

— псевдотипи.

Тригери визначаються як функції, що ініціюються DML-операціями. Наприклад, операція INSERT може запускати тригер, перевіряючий доданий запис на відповідності певним умовам. При написанні функцій для тригерів можуть використовуватися різні мови програмування. Тригери асоціюються з таблицями. Множинні тригери виконуються в алфавітному порядку [7].

3.1.4 Мова програмування C#

C# (вимовляється "See Sharp") – це проста, сучасна, об'єктно-орієнтована і строго типізована мова програмування. C# має коріння з сімейства мов C і є знайомою програмістам на C, C ++ та Java. C# стандартизована ECMA International як стандарт ECMA-334 та ISO / IEC як стандарт ISO / IEC 23270. Компілятор Microsoft C# для .NET Framework – це відповідна реалізація обох цих стандартів [16].

C# – об'єктно-орієнтована мова, але C# додатково включає підтримку компонентно орієнтованого програмування. Сучасний дизайн програмного забезпечення все більше покладається на програмні компоненти у вигляді автономних та самоописуючих пакетів функціональності. Ключовим для таких компонентів є те, що вони представляють модель програмування із властивостями, методами та подіями; вони мають атрибути, які надають декларативну інформацію про компонент; і вони включають власну документацію. C # надає мовні конструкції для повної підтримки цих концепцій, що робить C# дуже природною мовою, для якої можна створювати та використовувати програмні компоненти[17].

Деякі особливості C# (Рисунок 3.4) допомагають створювати надійні та довговічні програми.

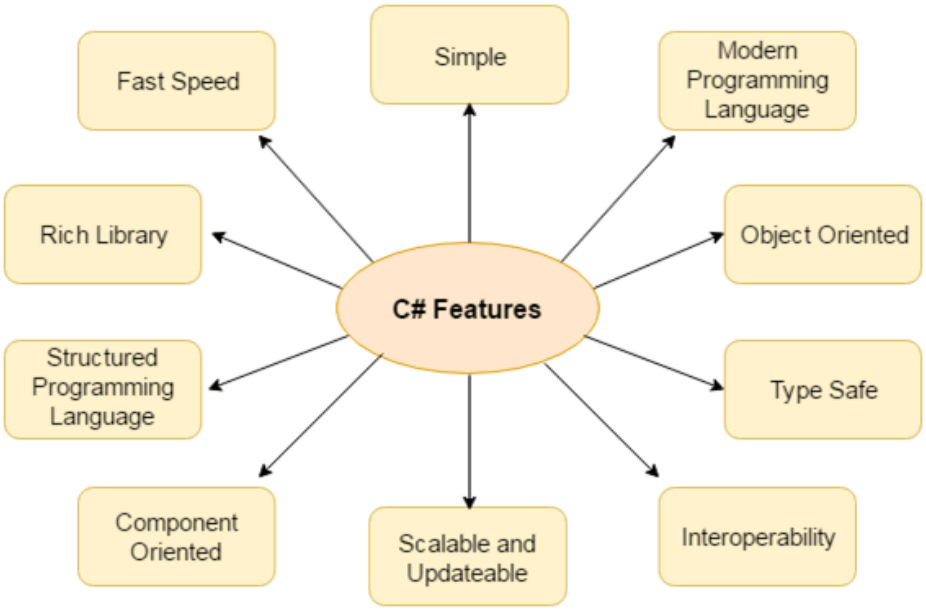

Рисунок 3.4 – Головні собливості мови C#

Збір сміття автоматично відновлює пам'ять, зайняту невикористаними об'єктами; обробка винятків забезпечує структурований та розширюваний підхід до виявлення та відновлення помилок; безпечна конструкція мови не дає змоги читати з неініціалізованих змінних, індексувати масиви за їх межами тощо.

C# має систему уніфікованих типів. Усі типи даних C#, включаючи примітивні типи, такі як int та double, успадковуються від єдиного типу об'єкту. Таким чином, усі типи розділяють набір загальних операцій, і значення будь-якого типу можуть зберігатися, транспортуватися та керуватися ними послідовно. Крім того, C# підтримує як визначені користувачем типи, що передаються за посиланням, так і типи, що передаються за значенням, що дозволяє динамічно розподілення об'єктів, а також вбудовуване зберігання легких структур[18].

Щоб переконатися, що програми та бібліотеки C # можуть розвиватися з часом, в розробці C# було зроблено великий акцент на версіях. Багато мов програмування приділяють цьому питанню мало уваги, і, як наслідок, програми, написані цими мовами, ламаються частіше, ніж потрібно, коли впроваджуються новіші версії залежних бібліотек. Аспекти дизайну C#, на які безпосередньо впливали можливі майбутні оновлення версій, включають окремі віртуальні модифікатори та модифікатори, що перевизначаються, правила перевантаження методів та інші особливості.

Ключовими поняттями в C# є програми, простори імен, типи, члени та збірки. Програми C# складаються з одного або декількох вихідних файлів. Програми декларують типи, які містять члени. Типи можуть бути організовані в простори імен. Класи та інтерфейси – приклади типів. Поля, методи, властивості та події – приклади членів. Коли програми C# компілюються, вони фізично упаковуються в збірки. Зазвичай збірки мають розширення .exe або .dll, залежно від того, реалізують вони програми чи бібліотеки.

Збірки містять виконуваний код у вигляді інструкцій Intermediate Language (IL) та символьну інформацію у вигляді метаданих. Перед його виконанням код IL в зборці автоматично перетворюється на специфічний для процесора код компілятором Just-In-Time (JIT) .NET Common Language Runtime.

.NET фреймворк відповідає за виконання програми C# в системі. Фреймворк – це поєднання мови програмування, що виконується, та набору бібліотек класів. Впровадження Common Language Infrastructure (CLI) здійснюється за допомогою Common Language Runtime (CLR).

Microsoft .NET Framework складається з таких основних компонентів:

— Common Language Runtime;

— бібліотека базових класів;

— користувацький інтерфейс програми;

Діаграма архітектури .NET фреймфорку разом із мовою програмування C# наведена нижче (Рисунок 3.5).

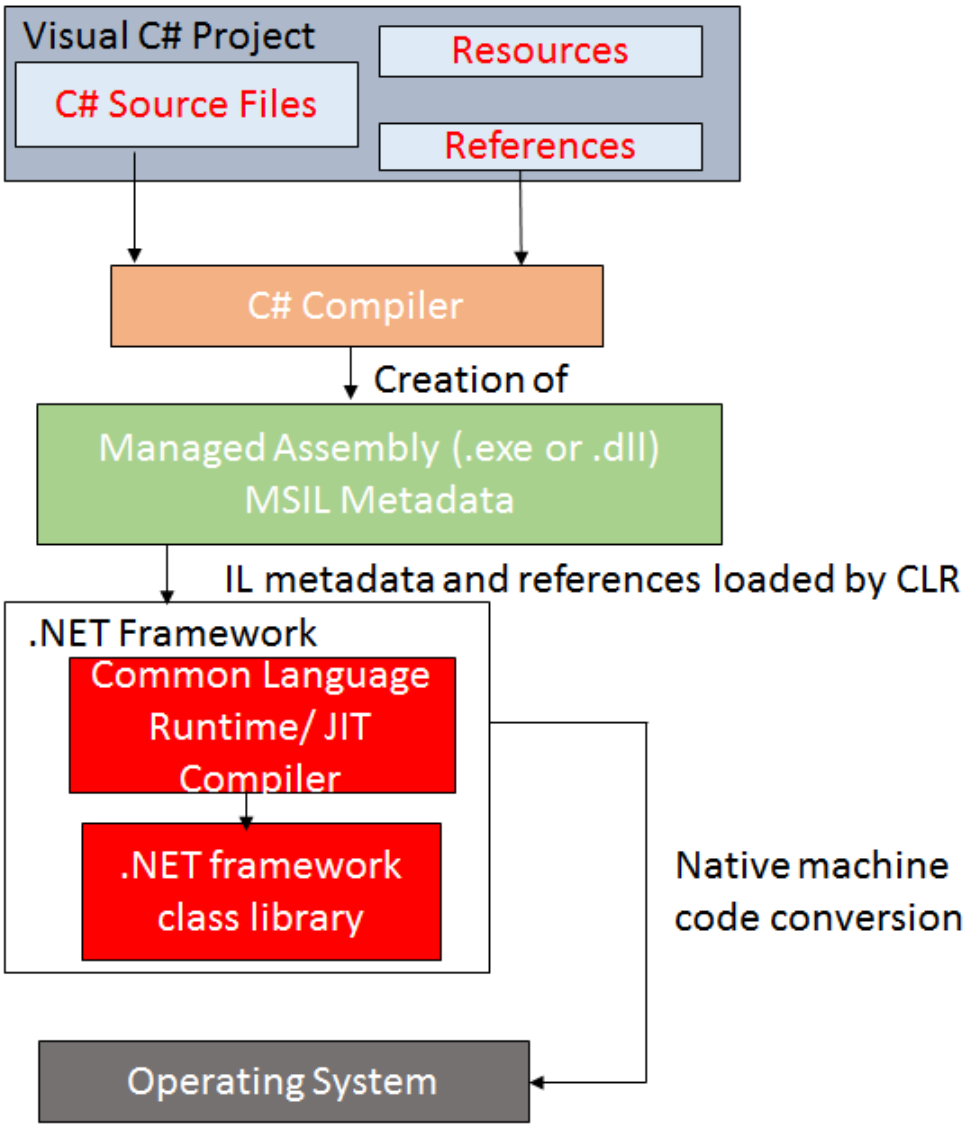

Рисунок 3.5 – Архітектура .NET фреймфорку

3.2 Архітектура програмної системи

Для зв’язку додатку із базою даних MySQL був використаний MySqlConnector.

MySqlConnector – постачальник даних ADO.NET для MySQL Server, MariaDB, Percona Server, Amazon Aurora, база даних Azure для MySQL, Google Cloud SQL для MySQL тощо. Він забезпечує реалізацію DbConnection, DbCommand, DbDataReader, DbTransaction – класи, необхідні для запиту та оновлення баз даних з керованого коду[13].

Перш ніж починати роботу с MySqlConnector необхідно в проекті додати посилання до нього (Рисунок 3.6).

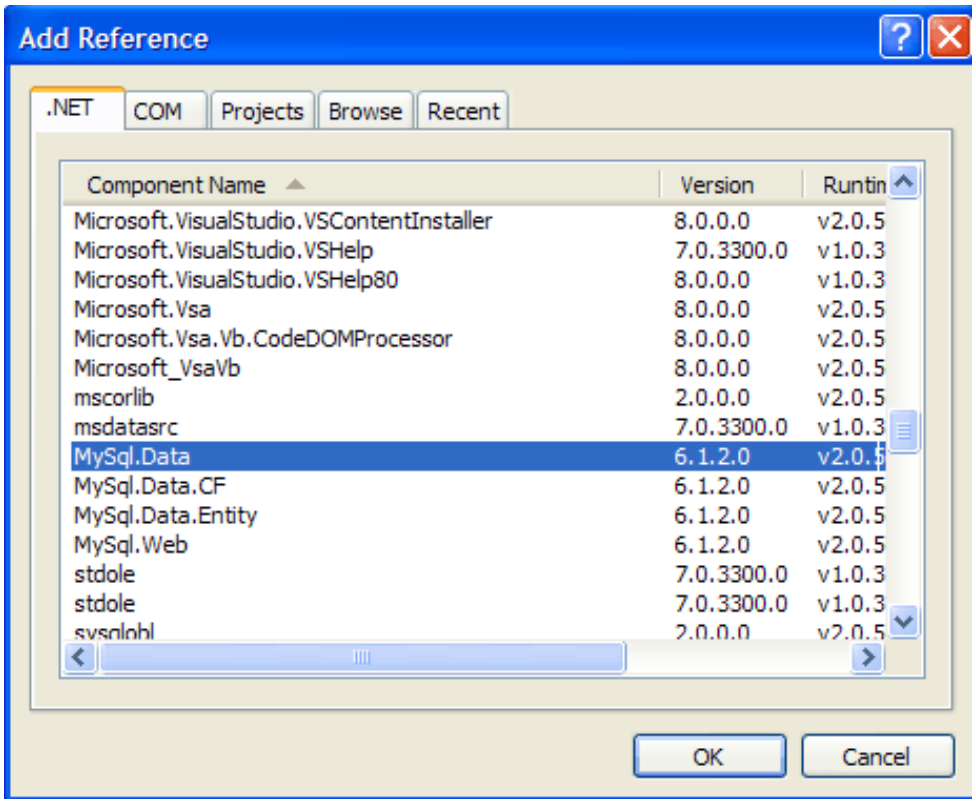

Рисунок 3.6 – Додавання MySQLConnector до проекту

Для роботи з MySQL необхідно у коді додати бібліотеку MySQL Connector: using MySql.Data.MySqlClient;

Для запитів і здійснення маніпуляцій за базами даних необхідно використовувати об’єкти класу MySqlConnection.

Вся взаємодія між .NET-додатком та сервером MySQL здійснюється шляхом об’єкта MySqlConnection при використанні класичного протоколу MySQL. Перш ніж програма може взаємодіяти з сервером, вона повинна створити об'єкт MySqlConnection, створити налаштування та відкрити MySqlConnection.

Для з’єднання з необхідною базою даних необхідно використати такі змінні:

— connection: буде використовуватися для відкриття з'єднання з базою даних;

— server: вказує, де розміщується наш сервер, у нашому випадку - localhost;

— database: це ім'я бази даних, яку ми будемо використовувати;

— uid: це наше ім'я користувача MySQL;

— password: це наш MySQL пароль;

— connectionString: містить рядок з'єднання для підключення до бази даних і буде призначений змінній з'єднання.

Для виконання оператора Select додаємо ще кілька кроків і використовуємо метод ExecuteReader, який поверне об’єкт dataReader для зчитування та зберігання даних або записів.

— відкриваємо з’єднання із базою даних;

— створюємо SQL команду;

— присвоюємо з’єднання і запит до команди. Це можна зробити за допомогою конструктора або за допомогою методів Connection та CommandText класу MySqlCommand;

— створюємо об’єкт MySqlDataReader для читання вибраних записів / даних;

— виконуємо команду;

— закриваємо об’єкт dataReader;

— закриваємо з'єднання.

Приклад коду одного зі з’єдань і виконання команди select із базою даних в даному проекті наведено нижче. (Рисунок 3.7).

```
string Connect = "Database=diplom;Data Source=localhost;User Id=root;Password=64nitoda";
string sql = "SELECT * FROM users WHERE  username='" + login.Text + "' and password='" + password.Text + "'";
MySqlConnection connection = new MySqlConnection(Connect);
MySqlCommand sqlCom = new MySqlCommand(sql, connection);
connection.Open();
sqlCom.ExecuteNonQuery();
MySqlDataAdapter dataAdapter = new MySqlDataAdapter(sqlCom);
DataTable dt = new DataTable();
dataAdapter.Fill(dt);
var myData = dt.Select();
if (dt.Rows.Count == 1)
```

Рисунок 3.7 – Код з’єднання з базою даних

Необхідно завжди відкривати з'єднання перед тим, як робити запит до нашої таблиці та закривати його відразу після закінчення роботи, щоб звільнити ресурси та вказати, що це з'єднання більше не потрібно.

Відкриття та закриття підключення до бази даних дуже просте, однак, завжди краще використовувати обробку винятків перед відкриттям з'єднання або закриттям, щоб виявити помилки та вирішити їх.

3.3 Опис бази даних

База даних складається з 4 таблиць: «users» (Користувачі), «number_of_attacks» (кількість атак), «db_attacks» (атаки на базу даних), «db_savings» (збереження баз даних).

Таблиця «users» призначена для зберігання даних про користувачів системи. В ній зберігається логін, пароль, ім’я користувача, роль(адмінісратор, користувач), посада, організація в якій працює користувач (Рисунок 3.8).

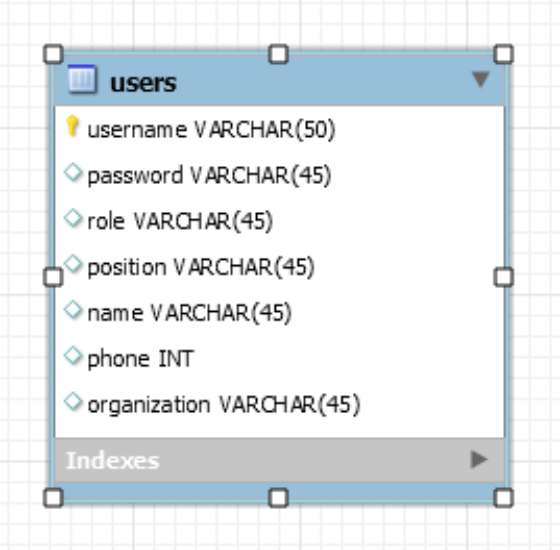

Рисунок 3.8 – Структура таблиці «users»

Таблиця «number_of_attacks» призначена для зберігання даних про атаки на систему. Зокрема, вона зберігає дані про час атаки, кількість атак, і порти на які було здійснено атаки (Рисунок 3.9).

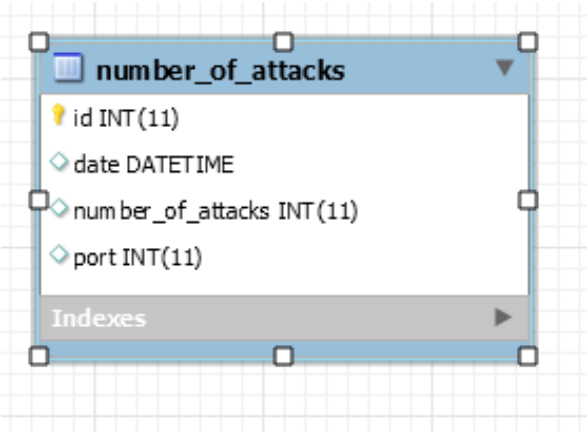

Рисунок 3.9 – Структура таблиці «number_of_attacks»

Таблиця “db_attacks” призначена для зберігання даних про атаки на бази даних. Зокрема, вона зберігає дані про час атак, їх кількість, а також імена баз даних на які була здійснена та чи інша атака (Рисунок 3.10).

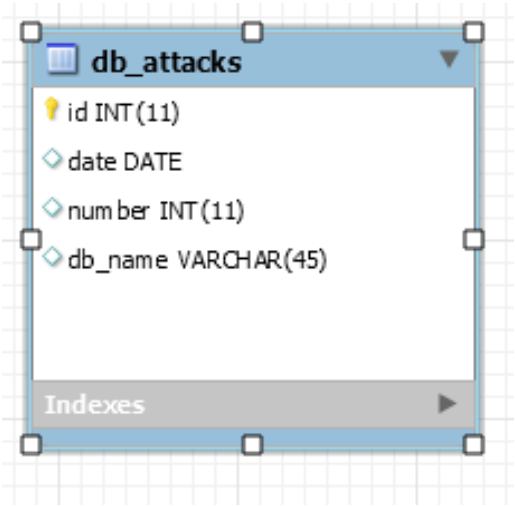

Рисунок 3.10 – Структура таблиці «db_attacks»

Таблиця «db_savings» призначена для зберігання інформації про історію збережень баз даних. Зокрема, про те, яку базу даних було збережно, дату збереження і хто виконував збереження (Рисунок 3.11).

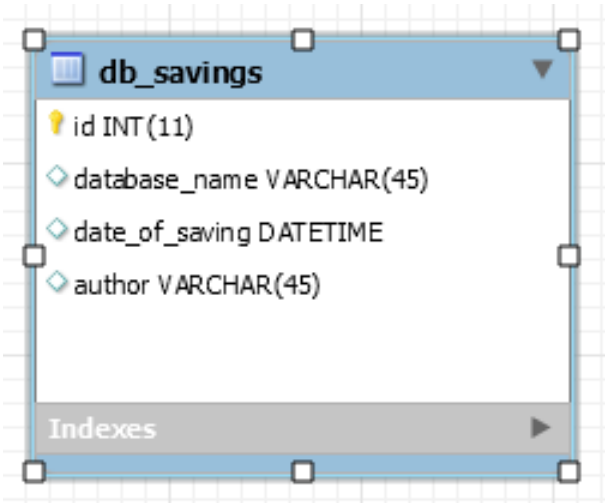

Рисунок 3.11 – Структура таблиці «db_savings»

На рисунку нижче також представлено приклад заповнення однієї з таблиць бази даних. (Рисунок 3.12).

Result Grid | Filter Rows: | Edit:

id	date	number_of_attacks	port
20	2019-10-07 02:56:38	29	7
21	2019-10-07 07:11:32	18	11
22	2019-10-07 12:50:26	5	7
23	2019-10-07 14:23:07	25	80
24	2019-10-07 14:37:08	28	80
25	2019-10-07 16:51:16	9	53
26	2019-10-07 22:09:00	37	53

Рисунок 3.12 – Приклад заповнення таблиці «number_of_attacks»

3.4 Висновки до розділу 3

В даному розділі було розглянуто основні інструменти для розробки системи підтримки приняття рішень при аналізі інформаційного забезпечення безпеки даних енергетичних процесів та систем.

C# було вибрано основною мовою для розробки системи, оскільки вона має необхідний набір перваг, якими має володіти сучасна мова програмування, і є оптимальним вибором для написання подібних систем підтримки прийняття рішень. Дана мова програмування має безліч переваг, такі як величезний набір бібліотек, які підключаються до проектів, що можуть полегшити розробку систем і також має модливості для розробки та проектування сучасного користувацького інтерфейсу. Було використано бібліотеку для побудови користувацького інтерфейсу ComponentOne. А також було використано бібліотеки для побудови зв’язку з базою даних для проведення маніпуляцій з таблицями.

Для розробки даної системи було необхідним провести дослідження можливих сучасних інструментів та засобів розробки для її створення. Це впливає на ефективність роботи при її розрбці, якість кінцевого програмного продукту, та його швидкість у роботі.

В якості СКБД було обрано систему MySQL, яка наразі є найбільш популярною та затребуваною реляційною СКБД. Вона має безліч зручних інструментів для

управління та підтримки баз даних, в тому числі і графічний інструмент MySQL Workbench.

При цьому було застосовано найбільш ефективний та швидкий спосіб для налагодження зв’язку між базою даних та додатком.

Для розробки і написання програмного продукту я обрала для використання інтегроване середовище розробки Visual Studio 2017, тому що воно надає максимальний набір інструментів для ефективного написання коду. Має можливості для автоматичного виправлення помилок при роботі, а також дає підказки під час написання коду, що дозволяє максимально швидко виконувати роботу. Для програмування на мові C# дане середовище є найоптимальнішим.

4. МЕТОДИКА РОБОТИ КОРИСТУВАЧА

У даному розділі описано системні вимоги до серверу для забезпечення стабільної та коректної роботи розробленої системи та опис роботи даної системи.

4.1 Інсталяція та системні вимоги

Розроблений програмний продукт, а саме його клієнтська частина працює на будь-якому персональному комп’ютері під управлінням операційної системи Windows. Рекомендовані вимоги до конфігурації ПК, який виступає в ролі сервера:

— операційна система Microsoft Windows 2000/XP/Server 2003/Vista;

— встановлена СКБД MySQL;

— налаштований вервер Apache;

— оперативна пам’ять 1024 Мб і вище;

— кількість вільного місця на жорсткому диску близько 150 Мбайт;

— роздільна здатність екрану 1280x800 пікселів;

— тактова частота процесору 1200 МГц;

— доступ до мережі Інтернет.

Рекомендовані вимоги до конфігурації пристрою, який виступає в ролі кінцевого споживача:

— оперативна пам'ять 512 Мб;

— тактова частота процесору 800 МГц;

— роздільна здатність екрану 1280x800 пікселів;

— кількість вільного місця на жорсткому диску 30 Мб;

— доступ до мережі Інтернет;

— операційна система Windows.

Для користування програмою на клієнтському комп’ютері потрібно відкрити її за допомогою ярлика програми. Для цього потрібно встановити програму на свій комп’ютер.

На комп’ютері, який виступає сервером, необхідно встановити MySQL Server для управління базою даних.

4.2 Сценарій роботи користувача з системою

Основна розробка програмного продукту поділялася на 2 частини: побудова інтерфейсу та бізнес-логіки програмного продукту та проектування та організація роботи бази даних.

Після запуску програмного застосунку користувач потрапляє на головну сторінку, де повинен пройти авторизацію для користування системою (Рисунок 4.1).

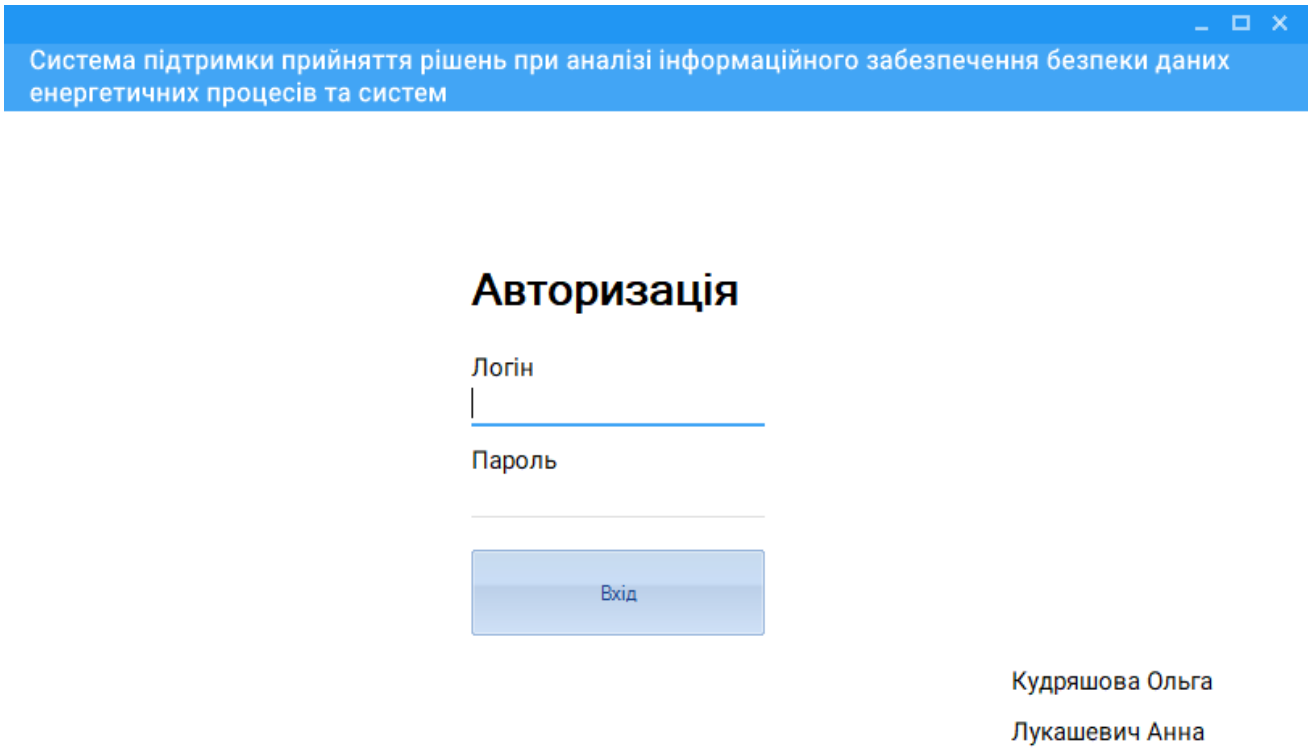

Рисунок 4.1 – Головна сторінка системи

При цьому при вході в систему передбачено валідацію введених даних, тобто будь-який користувач не зможе користуватися системою. (Рисунок 4.2).

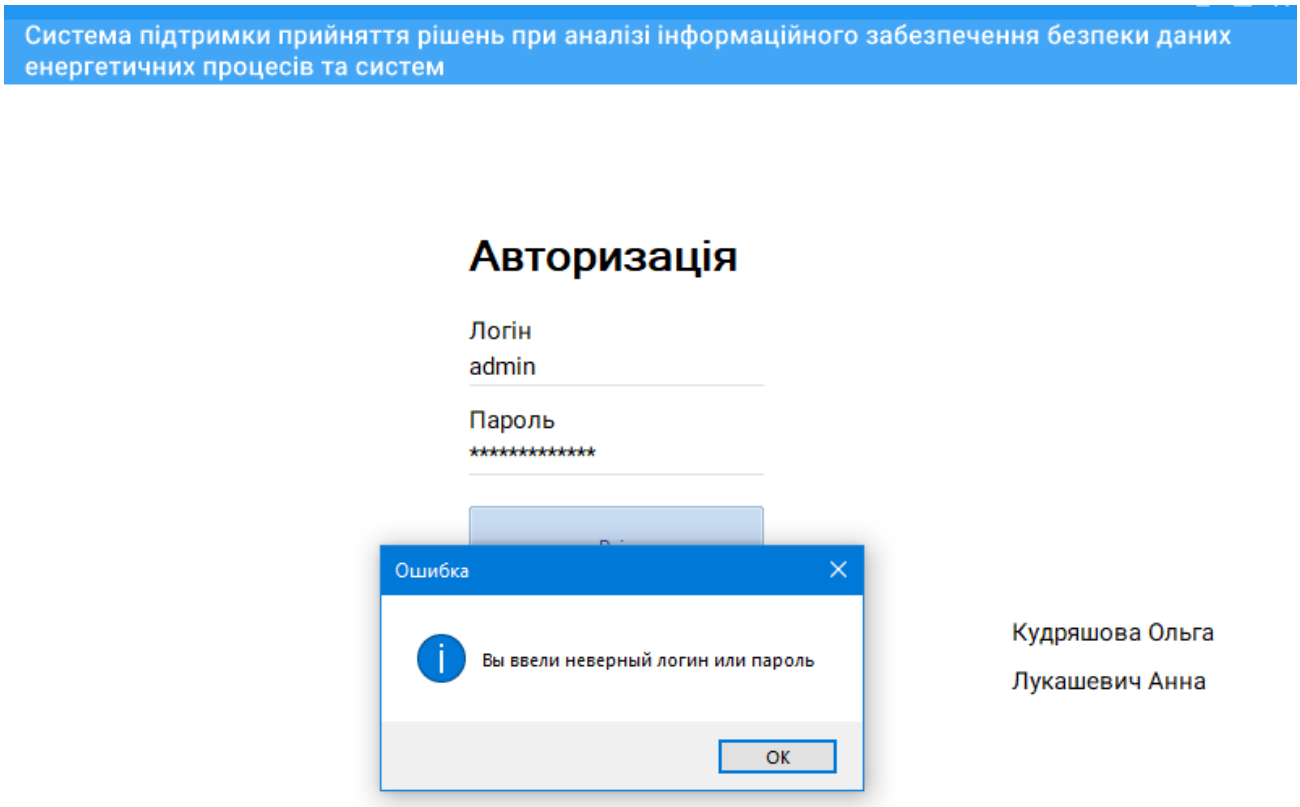

Рисунок 4.2 – Валідація даних при авторизації

Після входу жо системи для користувача відкривається головне вікно системи, яка складається з двох частин: підсистеми аналізу та підсистеми збереження даних (Рисунок 4.3).

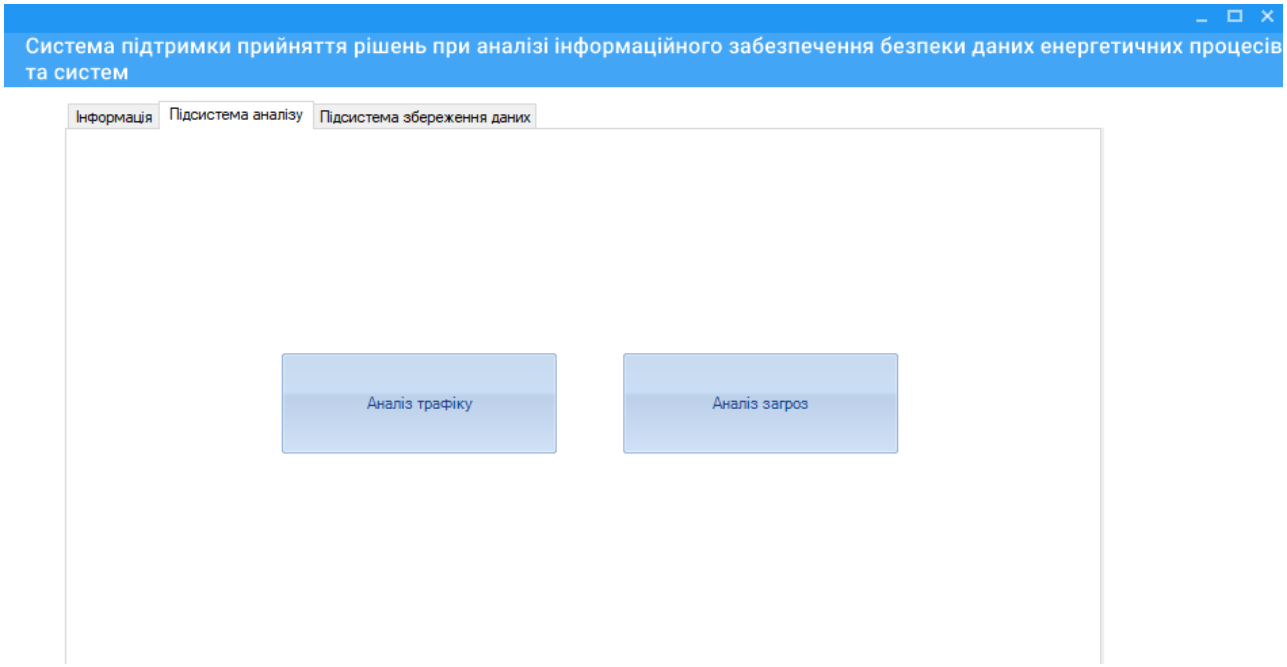

Рисунок 4.3 – Головне вікно системи

Користувач має натиснути вкладку «Підсистема збережння даних», щоб перейти до підсистеми збереження даних (Рисунок 4.4).

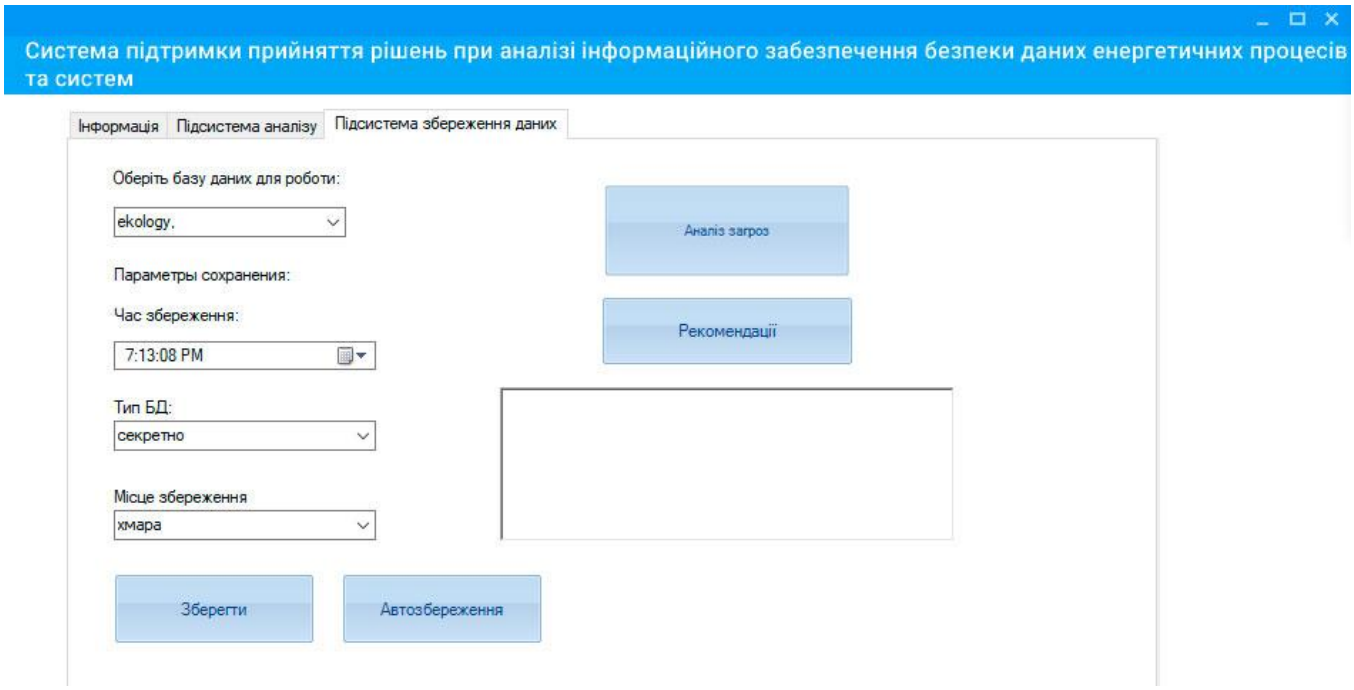

Рисунок 4.4 – Вікно підсистеми збереження даних

В даній підсистемі користувач має змогу зробити бекап (резервну копію) обраної бази даних з випадаючого списку. При цьому він має змогу обрати такі парметри зберігання як: тип бази даних (для службового користування, таємно, цілком таємно), місце збереження бази даних (архів, хмара, raid), а також обрати час для її збереження.

При цьому користувач має дві опції для збереження. Він може натиснути кнопку «Автозбереження», якщо хоче, щоб база даних зберігалася регулярно через певний проміжок часу. Або «Зберегти», якщо хоче зберегти базу даних єдиноразово.

Крім цього, користувач системи має змогу переглянути графіки, що демонструють спроби атак на систему, їх кількість в залежності від часу. Можливий перегляд графіків за тиждень (Рисунок 4.5) і за день (Рисунок 4.6). Зокрема, можливий перегляд даних і у табличному вигляді, оскільки для деяких користувачів такий спосіб є більш зручним та наочнішим.

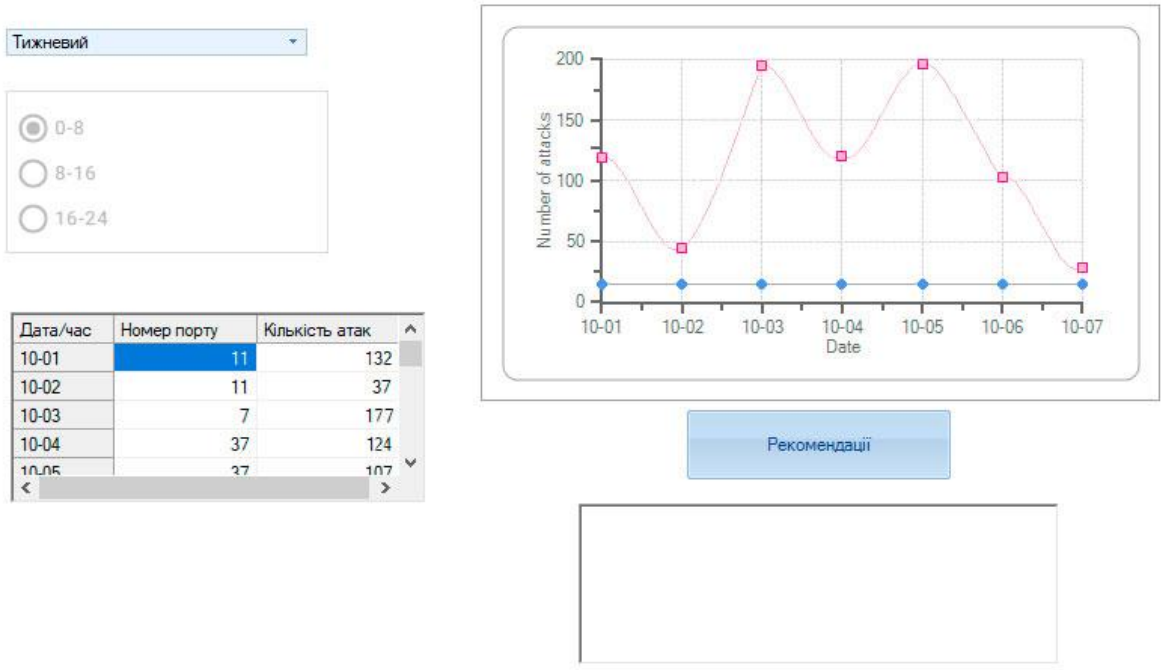

Рисунок 4.5 – Тижневий аналіз загроз для баз даних

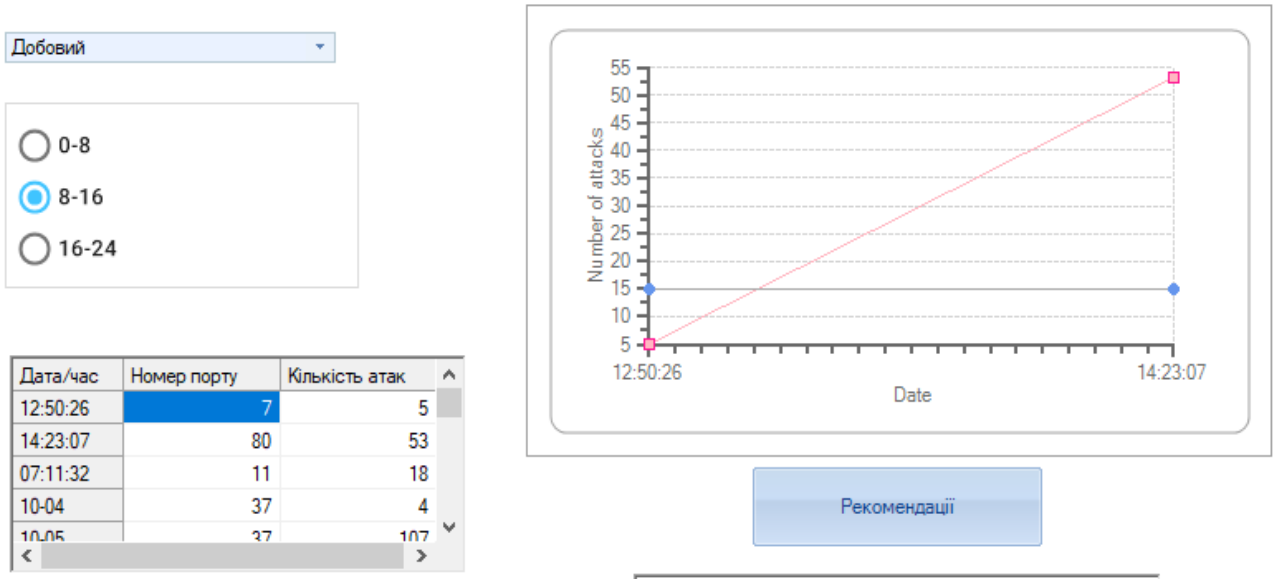

Рисунок 4.6 – Добовий аналіз загроз для баз даних

В залежності від власне проведеного аналізу користувач системи може отримати ряд рекомендацій для подальшого захисту даних та сисеми. Наприклад, зберігати дані більш часто або посилити контроль досутпу користувачів (Рисунок 4.7).

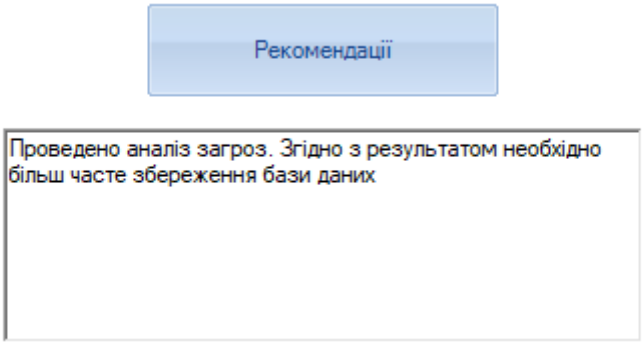

Рисунок 4.7 – Видані рекомендації користувачеві

Для більш комфортної роботи користувача з інтерфейсом було використано дві UI бібліотеки: Material Skin та C1 Component, що забезпечують сучаний Material дизайн у нашому додатку.

4.3 Висновки до розділу 4

У даному розділі представлено детальні інструкції як щодо інсталяції та системних вимог для встановлення додатку, так і щодо роботи їз системою підтримки прийняття рішень при аналізі інформаційного забезпечення безпеки даних енергетичних процесів та систем. Підсистема збереження даних.

Продемонстровано можливі сценарії роботи користувача з системою, показано зручність інтрефейсу користувача. А також, наприклад, описано детально, як працювати з системою та в деталях продемонстровано можливості підсистеми збереження даних.

5. СТАРТАП ПРОЕКТ

Розділ має за мету проведення маркетингового аналізу проекту як стартапу для аналізу та власне можливості його ринкового впровадження та можливих способівв та напрямів реалізації цього впровадження. Проведення маркетингового аналізу означає виконання певних визначених кроків та завдань.

5.1 Опис ідеї проекту

В даному підрозділі слід проаналізувати та подати у вигляді таблиць:

1. Зміст ідеї (що пропонується).
2. Можливі напрямки застосування.
3. Основні вигоди, що може отримати користувач товару.
4. Чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подаються у вигляді таблиці (таблиця 5.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки.

Таблиця 5.1. – Опис ідеї стартап-проекту

Зміст ідеї	**Напрямки застосування**	**Сегменти споживачів**
Розробка підсистеми збереження даних	1. Аналіз можливих атак на систему	Державні структури та компанії
	2. Видача рекомендацій користувачу щодо захисту системи	Приватні фірми, банківська сфера

Аналіз потенційних технолгічно-економічних переваг ідеї (різниця у порівнянні з конкуруючими приграмними рішеннями) порівняно із пропозиціями конкурентів передбачає:

1. Аналіз та ідентифікація всіх технологічно-економічних властивостей та характеристик ідеї.

2. Проведення дослідження та визначення потенційного кола конкурентів (програм-конкурентів) або товарів-замінників чи товарів-аналогів, що наразі існують на ринку, та проведення збору інформації та даних щодо значень технологічно-економічних показників щодо власного проекту та програм-конкурентів відповідно до зазначеного переліку.

3. Проведення порівняльного аналізу показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (таблиця 5.2).

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик

No п/п		**Потенційні програми/концепції конкурентів**		
		Мій проект	Система FortiDB	Система SafeNet
1	W слабка сторона	Відносно повільний час роботи	Повільний час роботи	Не є направленими на конкретну предметну область.
2	N нейтральна сторона	Можливіть роботи без доступу до інтернет	Занадто комплексна для простого користувача	Надто багато функцій
3	N нейтральна сторона	Невелика ресурсозатратність при роботі користувача	Потребує великої кількості ресурсів комп'тера для виконання	Потребує великої кількості ресурсів комп'тера для виконання

Таблиця 5.2 (Продовження)

4	S сильна сторона	Можливість засосовувати в установах різного спрямування	Можливість засосовувати в установах різного спрямування	Можливість засосовувати в установах різного спрямування
5	S сильна сторона	Своєчасне оновлення	Ширший функціонал	Ширший функціонал

5.2 Технологічний аудит ідеї проекту

В цьому підрозділі необхідно зробити аналіз технологій, за допомогою якоих можна реалізувати ідею проекту. Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 5.3):

1. За якою технологією буде виготовлено продукт згідно з ідеєю проекту.
2. Чи існують такі технології, чи їх потрібно розробити/доробити.
3. Чи доступні такі технології авторам проекту.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

No п/п	**Ідея проекту**	**Технології і реалізації**	**Наявність технологій**	**Доступність технологій**
1	Інтерфейс користувача	Мова програмування C#	Наявна	Умовна безкоштовно
2	База даних	СКБД MySQL	Наявна	Умовна безкоштовно

Таблиця 5.3 (продовження)

3	Алгоритм створення звітів	Мова програмування C#	Відсутня	Відсутня
4	Алторитм формування сценарію	Мова програмування C#	Відсутня	Відсутня
Висновок: проект реалізувати можливо				

За результатами аналізу таблиці можна зробити висновок щодо можливості технологічної реалізації проекту: так чи ні, а також технологічного шляху, яким цього варто досягти (з поміж названих технологій обираються такі, що доступні авторам проекту та є наявними на ринку).

5.3 Аналіз ринкових можливостей запуску стартапу

Передбачання і визначення ринкових можливостей, які можна використати під час ринкової імплементації проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дає змогу спланувати напрями розвитку проекту із урахуванням стану ринку, аналгічних пропозицій проектів-конкурентів.

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 5.4).

Таблиця 5.4. – Характеристика потенційного ринку стартап-проекту

No п/п	**Показники стану ринку (найменування)**	**Характеристика**
1	Кількість головних гравців, од	3
2	Загальний обсяг продажів, грн/ум.од	250
3	Динаміка ринку (якісна оцінка)	Зростає

Таблиця 5.4 (продовження)

4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	50

Середня норма рентабельності в галузі (або по ринку) порівнюється із банківським відсотком на вкладення. І якщо останній є вищим, можливо, є сенс вкласти кошти в інший проект.

За результатами аналізу таблиці робиться висновок щодо того, чи є ринок привабливим для входження за попереднью оцінкою.

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (таблиця 5.5).

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

No п/п	**Потреба, що формує ринок**	**Цільова аудиторія (цільові сегменти ринку)**	**Відмінності у поведінці різних потенційних цільових груп клієнтів**	**Вимоги споживачів до товару**
1	Надання підсистеми збереження даних, яку потребує ринок компаній, для яких важлива безпека даних	ЗВО, великі корпорації, банківські установи, науковці	Компанії заключають довготривалі договори, а стартапери віддають перевагу пробному терміну	стабільність роботи; невисока ціна; наявність випробувального періоду; наявність документації; підтримка необхідних платформ.

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиці 5.6-5.7).

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку. Аналіз пропозиції необхідно виконати аналізуючи існуючі види конкуренцій.

Необхідною умовою ефективного функціонування механізму саморегулювання ринкової економіки є конкуренція. Вона є важливою рушійною силою розвитку ринкової економічної системи. Конкуренцію породжують об'єктивні умови ринкового господарювання: різні форми власності на засоби виробництва, ринки збуту виробленої продукції, сфери використання капіталу з метою отримання найбільшого прибутку.

Конкуренція – це суперництво (змагальність) між різними учасниками ринкової економіки за найбільш вигідні умови виробництва та реалізації товарів і послуг, за привласнення найбільшого прибутку. Вона виступає силою, яка мобілізує особистий економічний інтерес і підприємницький потенціал та спрямована на їх максимальну реалізацію.

Захист конкуренції, суб'єктів господарювання і споживачів від недобросовісної конкуренції передбачає демонополізацію вітчизняної економіки і створення ринкового конкурентного середовища.

Таблиця 5.6 – Фактори загроз

No п/п	**Фактор**	**Зміст загрози**	**Можлива реакція компанії**
1	Підходить для нових проектів	Потребує визначеної структури бази даних	Імпорт схеми бази даних

Таблиця 5.6 (продовження)

2	Власний формат та вигдляд відображення ДС	При необхідності потрібно розробка сервісу преведення до визначенного формату	Додавання можливості автоматизованого експорту в різні типи сховищ, розробка додаткового ПЗ
3	Обмеженість функцій	Інструмент обмежений наявними функціями і не має деяких функцій, які мають конкуренти	Додавання нових функцій за потреби

Таблиця 5.7 – Фактори можливостей

No п/п	**Фактор**	**Зміст можливості**	**Можлива реакція компанії**
1	Незалежність від платформи	Можна використовувати на Linux, Windows, Mac операційних системах	Вихід на мобільний ринок, вихід на рівень web додатків
2	Недоліки в існуючих альтернативах	Можна використовувати на Linux, Windows, Mac операційних системах	Модифікація існуючих платформ

Аналіз пропозицій зображено на таблиці.

Таблиця 5.8. – Ступеневий аналіз конкуренції на ринку

Особливості конкурентної середовища	**В чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
1. Вказати тип конкуренції - монополія/олігополія/ монополістична/чиста	чиста	Прямі договори з стартапами, презентація продукту на виставках
2. За рівнем конкурентної боротьби - локальний/національний…	національний	Публікація статей на міжнародних сайтах
3. За галузевою ознакою – міжгалузева/ внутрішньогалузева	внутрішньогалузева	Розвивати напрямки систем формування сценаріїв
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	товарно-видова	Розповідати про свої переваги перед конкурентом у цій галузі
5. За характером конкурентних переваг - цінова / нецінова	нецінова	Надання функцій, які не надають конкуренти, оптимізація функцій, що мають конкуренти
6. За інтенсивністю - марочна/не марочна	марочна	Надання функцій, які не надають конкуренти, оптимізація функцій, що мають конкуренти

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (Таблиця 5.9).

Таблиця 5.9 – SWOT-аналіз проекту

Сильні сторони:	**Слабкі сторони:**
Актуальність користування системою, яка викликана бажанням забезпечення безпеки даних Оцінка проходить відразу для великої кількості людей, а також у будь-який період часу. Актуальність користування системою, що викликана постійним розвитком інформаційниз технологій, невелика ціна користування за місяць.	Потребує масштабної рекламної компанії Орієнтація на комп'ютерні додатки, які можуть відсіяти «не розвинутих» в технічному плані клієнтів Дороге зберігання великої кількості даних Обробка даних
Можливості:	**Загрози:**
Можливе продовження розробки проекту за кордоном, тому що проблема безпеки даних актуальна не лише в Україні Можливість створення звітів за виконаними аналізами Зручність у використанні	Відсутність користувачів через погану рекламну компанію Неможливість достукатися до необхідних API

5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 5.10).

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку.

Таблиця 5.10 – Цільові групи потенційних споживачів

No п/п	**Опис профілю цільової групи потенційних клієнтів**	**Готовність споживачів сприйняти продукт**	**Орієнтовний попит в межах цільової групи (сегменту)**	**Інтенсивність конкуренції в сегменті**	**Простота входу у сегмент**
1	Стартапери	Готові	Високий	Висока	Просто
2	Державні установи	Потребують недовгих переговорів	Високий	Середня	Складно
3	Ентерпрайз	Потребують довгих переговорів	Низький	Низька	Дуже складно
Висновок: як цільову групу обрано стартаперів					

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (таблиця 5.11).

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 5.11 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	**Стратегія охоплення ринку**	**Ключові конкурентос-проможні позиції відповідно до обрано їальтернативи**	**Базова стратегія розвитку***
Орієнтація поточної моделі на ринок стартаперів	Стратегія концентрованого маркетингу	Стартапери потребують швидкості розробки, яку надає підтримка декількох платформ даним продуктом	Стратегія спеціалізації (спирається на диференціацію)

Наступним кроком є вибір стратегії конкурентної поведінки (таблиця 5.12).

Таблиця 5.12 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	**Чи буде компанія шукати нових споживачів**	**Чи буде компанія копіювати основні характеристики конкурента**	**Стратегія конкурентної поведінки**
Ні	Шукати нових споживачів, забирати існуючих у конкурентів	Ні	Стратегія заняття конкурентної ніші

5.5 Аналіз ринкових можливостей запуску стартап-проекту

Для цього у таблиці 5.13 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.13 – Визначення ключових переваг концепції потенційного товару

Потреба	**Вигода, яку пропонує товар**	**Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)**
Пришвидшення оптимальності роботи алгоритму	Побудова оптимального формування сценарію за оптимальний час	Конкуренти або не мають орієнтованості на телекомунікацію, або формують сценарії не оптимальним шляхом

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання.

Після формування маркетингової моделі товару слід особливо відмітити чим саме проект буде захищено від копіювання.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (таблиця 5.14).

Таблиця 5.14 – Визначення меж встановлення ціни

No п/п	**Рівень цін на товари-замінники**	**Рівень цін на товари-аналоги**	**Рівень доходів цільової групи споживачів**	**Верхня та нижня межі встановлення ціни на товар/послугу**
1	5000…10000 грн	10000…13000 грн	50000…75000 грн	250…500 грн

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (таблиця 5.15):

1. Проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту).

2. Вибір та обґрунтування оптимальної глибини каналу збуту.

3. Вибір та обґрунтування виду посередників.

Таблиця 5.15 – Формування системи збуту

No п/п	**Специфіка закупівельної поведінки цільових клієнтів**	**Функції збуту, які має виконувати постачальник товару**	**Глибина каналу збуту**	**Оптимальна система збуту**
1	Клієнт повинен надаватися в режимах “тріал” та “повний” сплатити після закінчення випробувального сроку	Легість в встановленні, легкість в сплаті послуг	Веб-сайт	Проводити збут силами посередника формування сценаріїв

5.6 Висновки до розділу 5

Розроблений програмний продукт має ряд переваг над існуючими програмами-аналгами та є конкурентноздатним на ринку. Програма має шляхи подальшої розробки та розширення функціоналу, визначені маркетингові стратегії та шляхи збуту. Основна цільова аудиторія – працівники вузу та великі корпорації, для яких є важливим зберігання конфіденційних даних.

ВИСНОВКИ

У наш час все більш актуальними стають проблеми безпеки і збереженості конфіденційної інформації і даних. І тому дуже актуальною є проблема створення відповідного програмного забезпечення, яке разом із захистом даних буде спроможне допомагати приймати рішення кінцевому користувачеві.

У ході виконання даної магістерської дисертації було проведено аналіз існуючих систем, що вирішують подібні задачі (захист та збереження даних), та розроблено програмний засіб підтримки прийняття рішень при аналізі інфрмаційного забезпечення безпеки даних енергетичних процесів та систем. Підсистема збереження даних.

Головними перевагами розробленої системи є незалежність від операційної системи та простота використання. Система взаємодіє з СКБД, що дає можливість зберігати великі об’єми інформації і забезпечує швидкий пошук.

Розроблений програмний продукт має як певні переваги над аналогічними додатками, які представлені на ринку, так і ряд недоліків. До переваг можна віднести сучасний та легкий для сприйняття та роботи інтерфейс, забезпечення тільки необхідним набором функцій, швидкість роботи, дешевизна, порівняно з аналогами на ринку.

Враховуючи наведені вище вимоги до системи, програмне забезпечення було розроблено за допомогою технологій: системи керування базами даних MySQL, мови програмування C#, для побудови графічного інтерфейсу було використано систему Windows Forms, а для забезпечення додатку сучасним дизайном було додано бібліотеки MaterialSkin та ComponentOne.

Програмний продукт, що було розроблено у ході даної дипломної роботи, може стати ефективним механізмом , при малій кількості подібних систем. Користувачами системи можуть бути наукові співробітники, великі корпорації, науково-дослідні інститути.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Безопасность баз данных [Електронний ресурс] — Режим доступу: https://php.ru/manual/security.database.html

2. Основные аспекты безопасности СУБД: что следует знать [Електронний ресурс] — Режим доступу: https://tproger.ru/articles/db-security-basics/

3. Мельников В.П. Информационная безопасность и защита информации. / В.П.Мельников, С.А.Клейменов, А.М.Петраков // 3-е изд., стер. — М.: Академия, 2008. — 336 с.

4. Панасенко С.П. Комплексная защита информации. // Информационные технологии. -2001 - № 3 - с. 14-16.

5. Hassan A. Afyouni Database Security and Auditing: Protecting Data Integrity and Accessibility / Hassan A. Afyouni – Course Technology, 2012. — 429 pages.

6. Michael Gertz Handbook of Database Security: Applications and Trends / Michael Gertz, Sushil Jajodia — 1st edition, — Springer, 2008. — 592 pages.

7. Ron Ben Natan Implementing Database Security and Auditing / Ron Ben Natan — Digital Press, 2005 — 432 pages.

8. David Knox Oracle Database 12C Security / David Knox – McGraw-Hill Education, 2016 — 768 pages.

9. MySQL [Електронний ресурс] — Режим доступу: https://www.mysql.com/

10. Э. Гамма Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес — СПб: Питер, 2001. — 368 с.

11. Бойко В. В. Проектирование баз данных информационных систем / В. В. Бойко. — 2-ге вид. — М.: «Финансы и статистика», 1989. — 351 с.

12. Боровков А. И. Компьютерный инжиниринг / А. И. Боровков. — СПб: Изд-во Политехн. ун-та, 2012. — 93 с.

13. MySQL: Developer Zone [Електронний ресурс] — Режим доступу: https://dev.mysql.com/

14. Бэрон Шварц MySQL по максимуму. Оптимизация, репликация, резервное копирование / Б. Шварц, П. Зайцев, В. Ткаченко — СПб: Питер, 2001. — 864 с.

15. Database security issues [Електронний ресурс] — Режим доступу: https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2014/july/top-10-common-database-security-issues/

16. C# documentation [Електронний ресурс] — Режим доступу: https://docs.microsoft.com/en-us/dotnet/csharp/

17. Рихтер Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. / Рихтер Д. — СПб: Питер, 2018. — 896 с.

18. Петцольд Ч. Программирование для Microsoft Windows 8 / Петцольд Ч. — СПб: Питер, 2014. — 1008 с.

19. Безопасность SQL Server [Електронний ресурс] — Режим доступу: https://docs.microsoft.com/ru-ru/dotnet/framework/data/adonet/sql/sql-server-security